Université Paris XIII - Sorbonne Paris Nord
École Doctorale Sciences, Technologies, Santé Galilée

Métaapprentissage guidé par les connaissances du domaine

Metalearning guided by domain knowledge
in distributed and decentralized applications

THÈSE DE DOCTORAT
présentée par

Massinissa HAMIDI

pour l'obtention du grade de
DOCTEUR EN INFORMATIQUE

soutenue le 20 décembre 2022 devant le jury d'examen composé de :

Shanahan James	University of California Berkeley	Président du jury
Destercke Sébastien	Université de Technologie de Compiègne	Rapporteur
Read Jesse	École Polytechnique	Rapporteur
Amini Massih-Reza	Université Grenoble Alpes	Examinateur
Ghasemzadeh Hassan	Arizona State University	Examinateur
Pernelle Nathalie	Université Sorbonne Paris Nord	Examinatrice
Osmani Aomar	Université Sorbonne Paris Nord	Directeur de thèse

Résumé

Les catégories d'applications distribuées et décentralisées telles que l'internet des objets, l'industrie 4.0 ou la santé connectée imposent de nouveaux défis à la fois théoriques et pratiques pour l'apprentissage automatique. La dynamicité des déploiements, l'hétérogénéité des sources de données, la quantité de données disponibles, l'évolution des modèles de capteurs, les différences de perspectives, leurs chevauchements ainsi que leurs incohérences relatives sont autant d'éléments qui impactent fortement les performances des modèles d'apprentissage lorsqu'ils sont déployés dans le monde réel. Nous adoptons dans cette thèse le cadre du métaapprentissage et ses capacités à apprendre les biais inductifs appropriés. Contrairement aux modèles classiques qui fixent ces biais a priori, les modèles qui apprennent à apprendre offrent une flexibilité et un niveau de généralisation prometteurs pour pallier aux exigences du monde réel. Nous investiguons notamment l'apport des connaissances du domaine afin de guider le processus d'apprentissage : topologies des déploiements, lois de la physique, modèles analytiques, et dépendances entre les concepts à apprendre font partie des éléments incorporés explicitement dans le processus d'apprentissage. De nouveaux principes alliant, notamment, les représentations latentes universelles, les frontières de décision, les topologies des régions de classification ou la structuration des concepts à apprendre sont proposés.

Après avoir exposé le contexte applicatif et un état de l'art du méta-apprentissage et de l'apprentissage fédéré, nous présenterons les contributions qui s'articulent autour de trois axes. Nous proposons, dans un premier temps, deux nouvelles approches qui tirent parti des connaissances du domaine pour sélectionner et augmenter les exemples d'apprentissage. Les principaux problèmes traités dans cet axe sont l'hétérogénéité des sources de données et le coût des mesures effectuées par les capteurs et de leur transmission. Ensuite, nous proposons deux approches qui tirent parti de la sémantique de l'espace des labels (concepts à apprendre) afin de mieux organiser le processus d'apprentissage. L'idée est de décomposer le processus d'apprentissage en plusieurs sous-problèmes plus faciles à résoudre tout en maximisant la notion de réutilisation, de partage, et de transfert entre ces sous-problèmes. Enfin, nous nous concentrons sur les aspects collaboratifs des capteurs massivement distribués et sur les moyens d'améliorer la conciliation des apprenants décentralisés. Nous étudions des approches qui sont capables de fusionner efficacement les vues relatives fournies par les déploiements de capteurs, les abstraire de leurs biais contextuels et réconcilier les décisions prises par les apprenants décentralisés tout en tenant compte de leur relativité.

Toutes nos contributions sont validées par le développement d'approches pratiques évaluées sur des jeux de données provenant d'applications concrètes du monde réel telles que la reconnaissance d'activités humaine, le suivie du phénomène vibratoire dans les turbines industrielles et la reconnaissance des pleurs du nourrisson. Nos approches sont en mesure d'améliorer de manière significative les performances et la robustesse des modèles d'apprentissage sous des contraintes du monde réel, contribuant à surpasser les barrières menant au déploiement de tels modèles dans le monde réel.

Abstract

The category of distributed and decentralized applications, including the Internet of Things, Industry 4.0, and Connected Health, imposes new theoretical and practical challenges for machine learning. The dynamic nature of deployments, the heterogeneity of data sources, the amount of data available, the evolution of sensors models, the differences in perspectives, their overlaps, and their relative inconsistencies are all elements that have a substantial impact on the performance of learning models when deployed in the real world. We adopt in this thesis the framework of metalearning and its ability to learn appropriate inductive biases. Unlike classical models that fix these biases a priori, learning-to-learn models offer flexibility and a promising level of generalization to overcome the specificities of the real world. In particular, we investigate the contribution of domain knowledge in order to guide the learning process: topology of the deployments, laws of physics, analytical models, and dependencies between the concepts to learn are explicitly incorporated into the learning process. New principles combining universal latent representations, decision boundaries, the topology of the classification regions, or the structuring of concepts to learn are proposed.

After describing the applicative context and state of the art around meta-learning and federated learning, we will present our contributions which revolve around three axes. We first propose two novel approaches that leverage domain knowledge to select and augment learning examples. The main problems dealt with in this axis are the heterogeneity of the data sources and the cost of the measurements made by the sensors and their transmission. Then, we propose two approaches that take advantage of the semantics of the label space for organizing the learning process. The idea is to decompose the learning process into several sub-problems that are easier to solve while maximizing the notions of reuse, sharing, and transfer between these sub-problems. Finally, we focus on the collaborative aspects of the massively distributed sensing nodes and the ways the conciliation of decentralized learners can be improved. We investigate approaches that can efficiently fuse the relative views provided by the sensing environments, abstract them from their contextual bias, and conciliate the decisions taken by decentralized learners while considering their relativity.

All our contributions are validated by developing practical approaches evaluated on real-world datasets from diverse applications such as human activity recognition, monitoring the vibration phenomenon in industrial turbines, and infant cry recognition. Our approaches can significantly enhance the performance and robustness of learning models under real-world constraints, therefore contributing to lifting the limits for the deployments of learning models into the real world.

Contents

Chapter 1

Introduction

1.1 Motivation

With the ever-increasing quantities of sensing devices that surround every aspect of life, data is shifting from being fully centralized to massively distributed and decentralized. Current learning paradigms need to adapt to cope with the challenges that stem from this evolution. This is the case, for example, with Internet of Things (IoT) applications, which constitute very important societal, economic, and environmental challenges: smart sensors (data sources or data generators) equipped with ever-increasing computational capabilities are spreading very quickly and their adoption is only just beginning at the time of writing this thesis. Put simply, the Internet of Things is defined as an evolution of the Internet as we know it today, consisting in linking (everyday) objects endowed with perception, actuation, and computing capabilities, to the network of interconnections, the Internet. These capabilities then become accessible from everywhere.

The miniaturization tendency that characterizes today's sensor design is among the precursors and facilitating factors for the rapid development of pervasive technologies and their massive spread in many different areas. Sensing devices are becoming highly integrated, e.g., amplifiers, microcontrollers, radio chips, and antennas can be grouped into a unique and small surface (See Figure 1.1). This miniaturization facilitates the adoption of sensor deployments, e.g., wireless sensor networks, at large scales for monitoring and learning various phenomena. The ubiquity of these objects opens perspectives for developing a wide range of applications, such as smart homes, connected cars, smart health, and smart cities, to name a few. Sensing environments are key enablers driving, for example, the promotion of sustainable energy and health care delivery. In these environments, sensors (or data generators) are deployed on a massive scale following pre-defined structures to monitor industrial equipment, environmental factors, and ambient

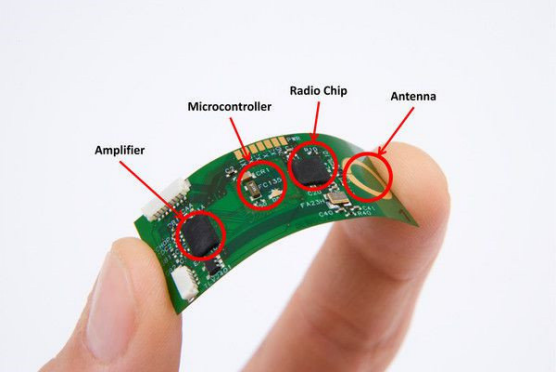


Figure 1.1: The miniaturization tendency of sensing devices pushes forward the adoption of IoT-enabled applications. Here is a flexible electrocardiograph ECG sensor with its components (amplifier, microcontroller, radio chip, and antenna) on a unique and small surface (courtesy of IMEC).

assisted living [Kim+22; Han+21; Lon+19].

The generalization and widespreadness of IoT devices' capabilities are accompanied nevertheless by some quite difficult challenges, especially when it comes to learning in the sensing environments they form. Beyond the classical difficulties faced by learning processes, other challenges stemming from the specificities of the sensing environments lead to significant impacts on performance. For example, the objects (or devices) that make up these sensing environments vary substantially in terms of their sensing characteristics, transmission models, and exact position in space. This leads, in particular, to differences in the data distributions across these objects and, ultimately, the inconsistency of the final learning objective. Additionally, the energy and computational constraints imposed on the individual sensing nodes shape the way the measurement process is performed (sampling frequency, quantization, etc.) and determine the availability of data. Furthermore, the entanglement between the cyber and physical domains, the relativity of the perspectives (or viewpoints) featured by the spatial disposition of the devices, and the dynamical factors of these environments make the sensing and learning configuration constantly evolve. They introduce a particular dimension of challenges that highlight the susceptibility of the learning process to domain specificities. Ethical and legal requirements, as well as the increasing need for model interpretability, are becoming increasingly prominent nowadays as machines take more significant roles in the process of decision-making, which is to the detriment of humans [Lip18; Gil+18; CPC19]. Because of the ubiquity, pervasiveness, and vulnerability of the connected objects, these requirements are even more exacerbated in distributed sensing environments. These are fundamental and widespread chal-

lenges that preclude the effective adoption of such applications. They need to be considered during the learning process and necessitate going beyond the classical paradigm of learning from data only.

Decentralized learning is a paradigm that naturally applies in this context. The distributed optimization setting was presented in [Kon+16] motivated by an increasingly spreading learning scenario consisting of a large number of mobile devices (also called clients) generating and holding training data locally instead of being aggregated into a unique centralized site. The goal is still to learn a unified theory while conciliating the diversity of clients in terms of the quantities of training data each individual client holds and the representativeness of the training samples of each client regarding the overall data distribution of the whole population. The general description of the federated learning setting was popularized by McMahan and Ramage in [MR17], while its theory was laid down in [Kon+16; McM+17]. Note that by the term *distributed*, we mean that the components of the system and the data they generate are in different places, and by *decentralized*, we mean that the decisions are not made by a single entity. Commonly, in decentralized systems, no node in the system is solely responsible for making decisions. Instead, it is the set of decisions at the node levels that leads a fortiori to a global decision. Decentralization imposes the notion of collaboration between the different nodes in order to achieve the overall objective.

In the decentralized learning paradigm, even if some of the proposed techniques handle issues related to distribution [McM+17], the heterogeneity of data distributions across clients [Hsi+17; Hsi+20], the aggregation from several sources [Lin+18], the inconsistency of computing capabilities across the components of the system [Wan+20b], transmission channel-awareness [Ren+20], and asynchronous communication constraints [Mis+18] to name a few, current learning approaches do not go as far as the IoT applications require. Still, these approaches are spread out and lack an integrated perspective that can handle all these aspects at once, i.e., an orchestrating entity (or mechanism) that can optimize all these aspects in a coordinated manner.

In parallel, tremendous efforts have been made to solve these challenges at different levels of the processing stack (or pipeline), either at the level of the learning process directly or at the local levels where these constraints emerge, e.g., sensing and processing, networking and transmission, or computing and scheduling levels. For example, regarding the variations in terms of the sensing characteristics across sensing devices, various approaches have been devised to temper the impact of heterogeneous components on the learning performances [Sti+15; KRM18; Yeo+21]. These strategies are efficient at solving very precise domain-specific problems but provide improvements at a particular level or to the aspects being considered at the expense of a joint optimization or reasoning, which would provide further improve-

ments. Indeed, it ends up that all these constraints are tightly linked together, and the interplay between them and, in particular, with the learning process, is significant. It is, therefore, not suitable to continue treating each of these aspects in isolation as they often impact each other.

This thesis investigates how to learn efficiently in the context of generalization and the widespreadness of sensing, actuation, and computing capabilities, with all domain specificities and constraints surrounding it. More precisely, we propose to build a unifying framework where all the aforementioned domain specificities and constraints are expressed in suitable forms and integrated into the learning pipeline. The idea is to be able to reason about the interplay between these aspects and optimize their corresponding learning mechanisms in the learning pipeline. Ultimately, the goal is to improve the performance of the learning processes while complying with the various requirements and constraints, e.g., ethical and legal requirements and physical constraints. At first, it may seem that integrating all these aspects and treating them at once will lead to further complexifying the problems at hand. Basically, one seeks to decompose the problems into small subproblems that can be easily solved. We are not contradicting this principle. However, we aim to handle these problems at a higher level of processing, which has the potential to lead to a "bigger picture" of the seemingly isolated but highly correlated and interacting problems. This idea is rooted in rather philosophical considerations related to the notion of complexity that characterizes the multi-dimensional interplay between the components of a given system or model at a local level and leading ultimately to the emergence of properties that the components taken individually do not posses [Joh02]. As Juignet frames the philosophical concept of "complex thinking" presented by Morin in [Mor15; Mor07]: "Complexity requires trying to understand the relationships between the whole and the parts. But knowledge of the parts is not enough for knowledge of the whole; we have to go back and forth in a loop to bring together the knowledge of the whole and that of the parts."

This led us to pursue two complementary axes. On the one hand, studying regularities that emerge across problems (and more broadly, across learning tasks, environmental conditions, situations, sensor configurations, sensing characteristics, etc.) and ways to leverage past experience gained on past tasks in order to improve the way we deal with current ones. For this, we build upon the learning-to-learn framework [Sch87; BBC91; Fin18], which, unlike classical models that fix these biases a priori, offers flexibility and a promising level of generalization to overcome the specificities of the real world. These models basically look at how to put the learning process in appropriate conditions to pursue the learning. This involves exhibiting hyperparameters of the learning process—that is, the underlying learning mechanisms like the architecture of a neural network or the preprocessing steps

applied to the inputs—that can be acted upon based on past learning experience (or knowledge about the learning itself) in order to improve the performance of the learning process at the current learning task or future never-seen tasks. On the other hand, modeling (or expressing/representing) the various domain-specific requirements and constraints, as well as the interplay between the different parts of the systems using appropriate tools, and then finding the correspondence between these domain specificities and the learner's hyperparameters (exhibited above) so that these aspects can be optimized together with the learning process. These are further detailed in Section 1.3.

In the rest of this chapter, we will discuss in more detail the precise research questions arising from these kinds of environments and summarize the contributions and broader impact of the thesis.

1.2 Domain-specific requirements and constraints

In this section, we enumerate the problems arising in distributed sensing environments, exemplified by concrete real-world applications. A detailed overview of these problems can be found in one of our previous contribution [HO21b]. These problems form the core components of the research questions that we are aiming to answer in this thesis. If familiar with these aspects, the reader can skip this section and go straight to the research questions summarized in Section 1.2.1.

Characteristics of the sensing devices and induced data heterogeneity

Data acquisition is determined by various factors, including the intrinsic characteristics of the sensors, which depend on the different components involved in transforming the sensed phenomenon into an electrical signal (See Figure 1.2). Each of these components exhibits, in turn, various characteristics related to their designs which ultimately shape the resulting electrical signal. In particular, as depicted in Figure 1.3, the design of analog-to-digital (ADC) converters obeys a trade-off involving simultaneously conversion accuracy, transformation speed, and power, which leads ultimately to mitigating the overall sensor's performance.

The performance characteristics of a sensor are just as important as its basic function, which is to detect and gauge the phenomenon of interest [Ida14]. In addition to the type of sensing modality, the choice of an appropriate sensing device and its performance characteristics for a given application is one of the most important issues distributed sensing environment designers are faced with. The transfer function defines the relation between the input of the sensing device and its output. Depending on many different factors, sensing characteristics defined by this

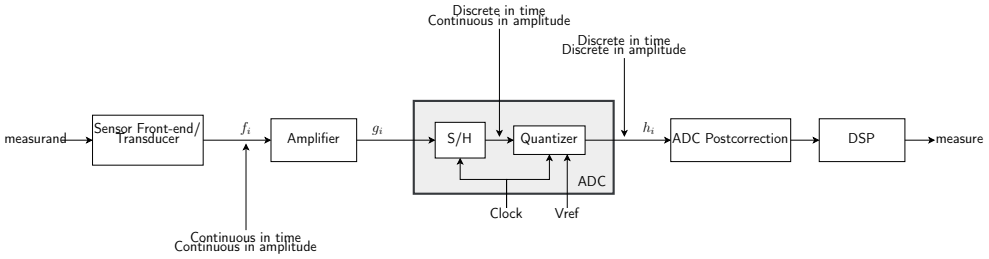


Figure 1.2: The measurement of a phenomenon as simple as temperature through a sensor is in itself an inductive process involving many biases. The action of the physico-electrical process of the sensor generates an electrical signal proportional to the physical phenomenon being measured. We, actually, do not have access to the physical phenomenon itself but to a representation provided through a transfer function deduced mathematically and which is specific to the physico-electrical process of the sensor. The choice of this process constitutes a bias similarly to the elaboration of the transfer function.

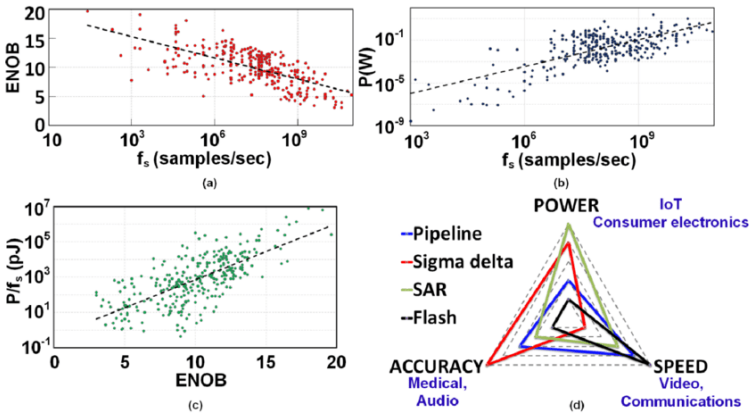


Figure 1.3: Trade-offs in conventional analog-to-digital architectures between (a) speed and accuracy, (b) speed and power, (c) accuracy and energy, as reported in [Bor]. (d) Spider diagram of analog-to-digital architectures (different color lines), design trade-off, and associated applications (in blue). (from [Dan+18]).

transfer function may vary substantially. Besides, depending on the application or the phenomenon being monitored, many different properties are considered with varying importance by the designers, including *span*, *accuracy*, *frequency response*,

sensitivity, *repeatability*, *resolution*, and *reliability*. Other factors such as the cost are also considered. In particular, in the case of mobile computing and applications based on the use of smartphones, the considered sensors are often low-cost leading on many occasions to poor calibration, inaccuracies, and limitations in the granularity and range compared to using dedicated inertial measurement units [Tru+13; Sti+15; Sti+15; CD15]. For example, in the specific case of accelerometer sensors, a number of important factors are evaluated in [Alb+08]. The important ones are the *sensitivity*, which defines the ratio of its electrical output to its mechanical input, the *amplitude limit* specifying the maximum range of acceleration that can be measured, the *shock limit*, the *natural frequency*, the *resolution*, the *frequency range*, and the *phase shift* defining the time delay between the mechanical input and the corresponding electrical output signal of the instrumentation system. The

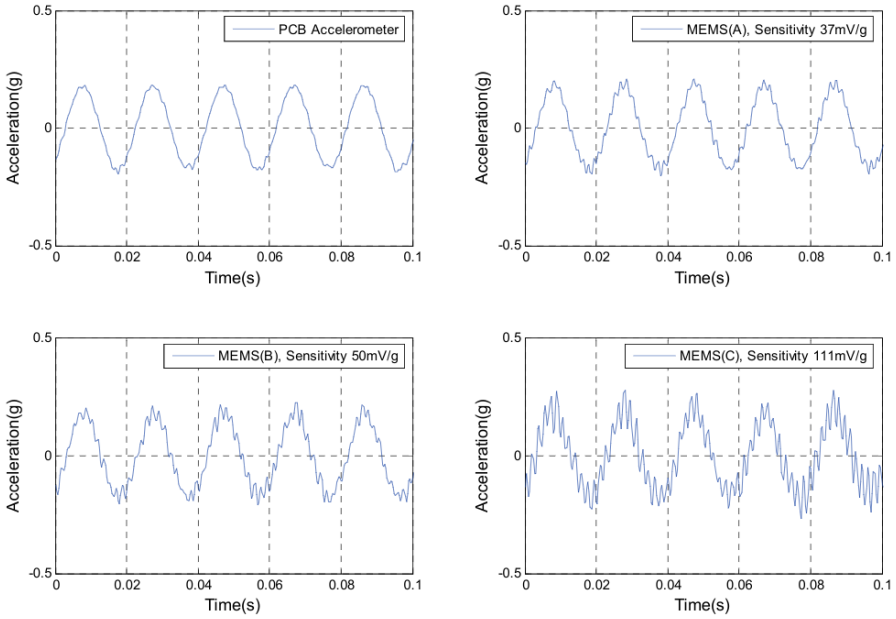


Figure 1.4: A concrete example of how varying device characteristics lead to substantial differences in the sensor outputs. From [Alb+08]: Here are shown the measured acceleration responses by different MEMS accelerometers (A, B, and C) and the reference (PCB) accelerometer at 53Hz for the excitation amplitude 0.15g.

responses generated by four MEMS accelerometers with various characteristics were investigated according to the aforementioned factors. Figure 1.4 summarizes the measured acceleration responses with various characteristics compared to a reference accelerometer and indicate that, in some configurations, there is a lot of noise, including extra un-interpretable peaks when compared against the reference

accelerometer and against the remaining ones. A substantial noise and shift in phase are also observed.

These characteristics were also investigated regarding their tangible impact on various applications, noticeably human activity recognition (HAR) from sensor-enabled smartphones. Device-instance diversity, i.e., variations in the sensor observations of the same phenomenon across different device instances, and its systematic impact on the learning process is another form of diversity exhibited in real-world applications (e.g., human activity recognition [Dey+14; Jan+17; KRM18], autonomous vehicles [Yeo+21]). For example, authors in [Sti+15] investigated in a systematic manner sensor-, device- and workload-specific heterogeneities using smartphones and smartwatches, consisting of different device models from various manufacturers. Beyond the obtained results indicating that these heterogeneities significantly impair the performances of activity recognition models, this leads to asking a set of research questions noticeably: "How characteristics of sensing devices impact the learning process (heterogeneity, distribution skew, etc.)? How to account for the characteristics of the sensing devices and their evolution during deployment in real-life settings?"

Some approaches in the literature are often focused on grouping the devices based on their characteristics prior to the learning process. Indeed, to mitigate the impact of device heterogeneity on machine learning models (HAR, specifically), Stisen et al., for example, proposed an approach that, first, clusters the devices based on their characteristics, then builds a model for each obtained cluster. In the same vein, other approaches indicate the advantages of performing such prior modeling on the learning pipeline. Even if this is clearly an example of leveraging prior knowledge to mitigate the impact of device diversity, these kinds of approaches are limited to the construction personalized (or cluster-specific) models and do not consider the collective dimension of the sensing environments to learn unified theories. This being said, there exist potential avenues for integrating additional knowledge about the characteristics of the sensing devices and their evolution for learning such unified theories. This ultimately can alleviate the need for building specific models for devices exhibiting similar heterogeneities.

Physical constraints

Domain-specific constraints, including energy, transmission, and computational constraints, can be inherent to the deployed end-devices, imposed by regulatory requirements, or related to contextual and environmental factors.

Among the existing domain specificities, the energetic autonomy of the deployed end devices is probably one of the most important, as it also has an impact on the transmission and computational aspects of the entire deployment. Indeed, it results in numerous restrictions imposed on the operations of the deployment

and ultimately on the learning process. Autonomy generates trade-offs involving the capacity of the nodes to sense and monitor the phenomenon being considered. Energy constraints influence the computing capacities of the end devices and have direct consequences on the data sampling frequency, transmission frequency, and local signal processing pipeline.

The impact that the energy constraints impose on the learning process is significant. For example, during the conciliation step performed in the decentralized learning setting, the heterogeneity in terms of computational capabilities exhibited by the distributed end devices leads to a major problem, which is learning objective inconsistency. A problem studied, for example, in [Wan+20b]. Basically, when running the decentralized model's updates locally, the local nodes exhibiting heterogeneous computational capabilities lead to diverging local models.

Various approaches have been proposed in the literature to account for these energy constraints. For example, authors in [MAL12] presented an energy-efficient, thermal- and power-aware routing algorithm for on-body sensor deployments which considers the node's temperature, energy level, and received power from adjacent nodes in the objective function design. Even by increasing battery capacities and optimizing hardware components and signal processing pipelines, e.g., the development of low-power hardware designs for the architectures, processors, or transceivers [SK10; Bha+20], the problem is only shifted.

Furthermore, the spatial disposition of sensors and their transmissions are subject to dynamical factors such as path loss and the vulnerability of the radio channel, which is used to connect, in a wireless fashion, the various on-body sensors together. This radio channel is impacted by noise and interference, which, in addition, evolve with time as a result, in the case of on-body sensor deployments, of the body movements and the environment (e.g., reflections of the radio waves on the walls) leading eventually to path loss and the impossibility to transmit data [Gol05]. Figure 1.5 illustrates the impact of the transceivers' on-body locations on the path loss. Furthermore, authors in [Gor+09] studied the problem of path loss with respect to the underlying network topology, noticeably star vs. multi-hope mesh, where a reduction of the emitter-receiver distance could counteract this problem. In addition to the impact of the on-body sensors placement on the path loss, the body movements as well as the surrounding environment have a big influence on signal propagation and subsequently on the packets transmissions. Authors in [For+05], for example, studied the influence of arm motions, while authors in [DO09] considered the impact of various types of activities (still, walking, and running) on the path loss depending on the location of the transceivers. Similarly, the impact of the surrounding environment has been investigated by authors in [For+05] who studied signal propagation by taking into account factors related to the environment in which the user operates. These include, for example, the

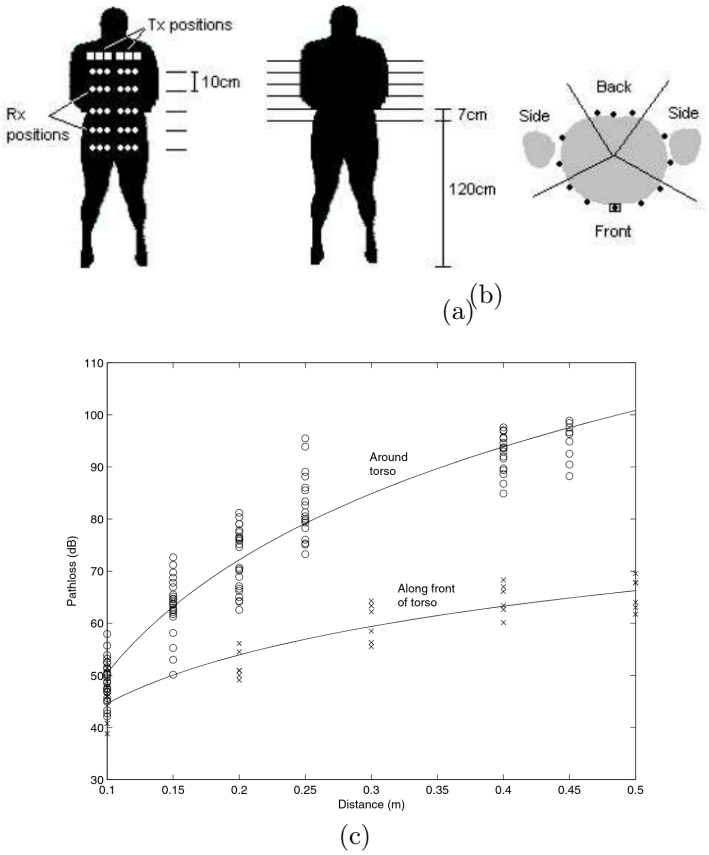


Figure 1.5: Illustration of the impact that physical constraints and environmental factors impose on the data transmissions (here the path loss). Figure from [For+05]: on-body placement of the sensing nodes (a) along the torso and (b) around the torso. (c) Measurement of the path loss (dB) as a function of the distance (m) between the sensing nodes around the torso (top line) and along the torso (bottom line).

influence of ground reflections, considered more reliable to be exploited during transmission, as well as reflections from surrounding environments on received signals.

Due to these constraints, transmissions have to be performed sparsingly. Again, in the decentralized learning setting, the necessary updates and conciliation phases performed between the local devices and the central parameters server are thus impacted, which could eventually lead the central server to completely ignore the updates (or contributions) made by a given device or group of devices. Scheduling the most informative local learning updates by taking into account transmission chan-

nel diversity is one avenue that has been pursued in [Ren+20]. And in [Rau+17], authors investigate the selection of network interfaces, where the radio used to transmit is selected depending on the environment opportunities (bandwidth, link quality, energy).

Therefore, the questions that we can ask here are: "How to account for the heavy cost of performing sensing operations when facing energy and computational constraints? How representative are the end results of sensing operations w.r.t. the example space? Can we leverage prior knowledge about, e.g., the phenomena being monitored and how they propagate, in order to reinforce representativeness of the example space?" Furthermore, "can we organize the learning process in a way that accounts for the heterogeneous computational capabilities? How do energy and computational constraints shape data acquisition and transmissions, which ultimately impact data quality as well as data availability (the problem of partial or incomplete views)? And how to account for the constraints imposed on transmitting data over networks (e.g., energy and computational constraints, limited bandwidth, path loss, etc.)?"

Entanglement between the cyber and physical domains

The notion of intrication between the cyber and physical worlds can be illustrated by the single-sensor deployment depicted in Figure 1.6. We can see in this figure the tight link that exists between the sensing devices (the cyber-dimension, in this case, the optical sensor) and the entity being monitored (the physical dimension, in this case, the human body via its skin).

This entanglement is of utmost importance and has a substantial impact on the subsequent signal analyses and learning process. For example, one of the problems that raises here is related to the heat generated by the sensors, which sometimes modify the collected data by increasing, e.g., the temperature of the body. More concretely, in the case of a fingertip pulse oximeter, the author in [Ban12] points out the substantial effects of the dissipated heat from the device on the human body temperature depending on the sampling frequency. Similarly, mobility, physiological condition, mood, and time of the day are additional contextual factors that affect human body parameters and, ultimately, the data acquisition process [Ban12]. For example, in the case of the photoplethysmograph (PPG), which measures the heart rate via the variations of intensity of reflected fraction LED-emitted light (see Figure 1.7(a)), the measured PPG signals are found in [Anz+20] to be extremely vulnerable to body movements: body movements affect the shape of blood vessels and surrounding tissues which lead to low-accuracy measurements, which are further accentuated by the low power constraints (see Figure 1.7(b)).

One can notice how entangled the sensing environments that we are dealing with. The amount of components that interact in a seemingly simple learning

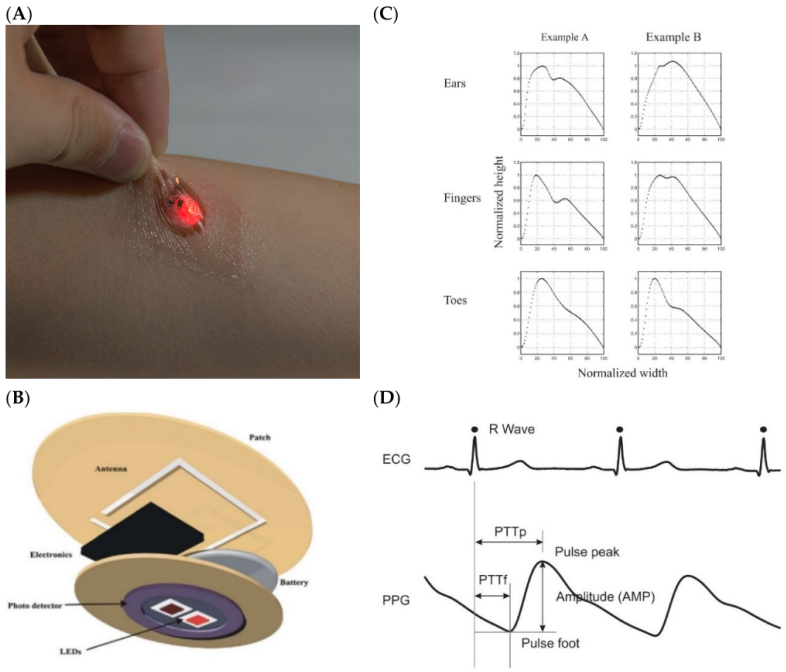


Figure 1.6: Illustration of the entanglement between the cyber and physical domains and how the interplay between various aspects of the deployments leads to impairments in the end results of the sensing processes. From [AMK20]: here are depicted (a) an optical sensor mounted on skin, (b) a basic diagram of a wireless photoplethysmography (PPG) sensor, (c) a PPG signal collected from three different locations, (d) a comparison of the PPG signal with ECG signal.

task like temperature monitoring is tremendously important, likewise the impact that all these aspects have on the learning process. The entanglement can take different forms, e.g., the impact of the sensing devices on the phenomena being monitored or the impact of various contextual elements like the current mood on the modality being analyzed. This translates perfectly the concept of "complexity" mentioned previously and leads to the research questions: "How to account for the entanglement between the cyber and physical domains and its impact on the sensing process? Furthermore, how to organize computations (learning problems and sub-problems) in order to account for the energy and computational constraints of the deployment-end computing devices? how to account for the impact of the computing devices on the phenomena being monitored? how to account for the various contextual elements that alter the considered learning problems?"

Various languages and formalisms have been devised in order to model this entanglement and capture the mutual effects between the physical units used to

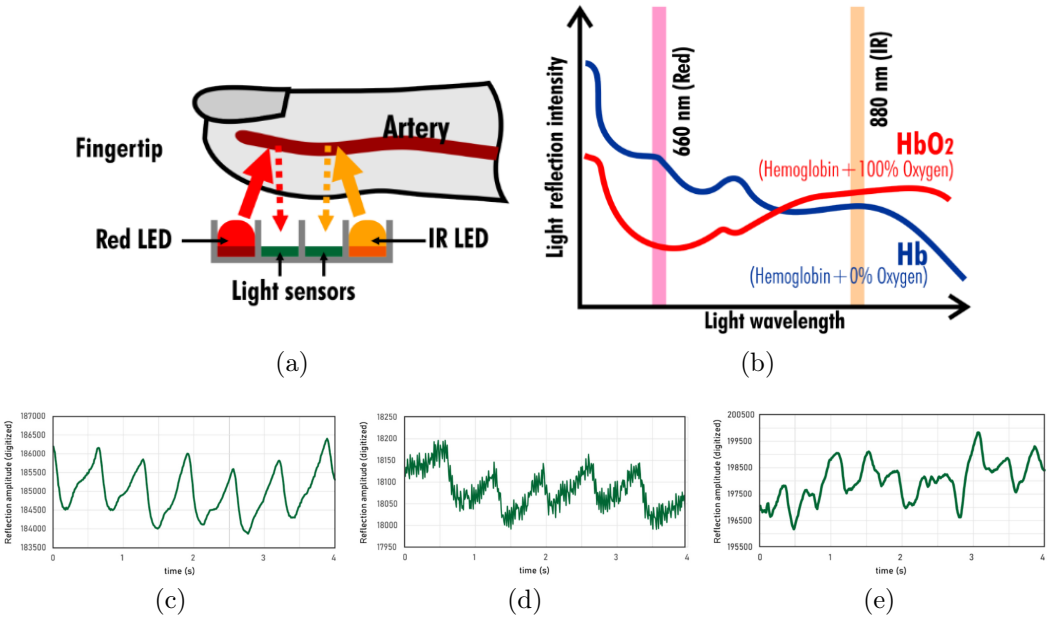


Figure 1.7: Another example of the entanglement between the cyber and physical domains. From [Anz+20]: (a) Disposition of the physical sensing device (photoplethysmograph) on the area of interest and (b) an example of the signal being generated. (bottom) Illustration of the PPG signal quality samples collected from a fingertip in different configurations (electric current/physical activity): (c) high current, sleeping, (d) low current, sleeping, and (e) high current, running.

sense a region of interest and the physical property of interest. For example, the architecture analysis and design language (AADL) [FGH06] is an industry-standard specifically addressed to mission- and safety-critical systems where the physical system, computer hardware, software, and their interactions are expressed using textual and graphical notation. Similarly, Banerjee in his thesis proposed abstract modeling of the cyber-physical systems, which takes into account the intentional and unintentional interactions between the cyber components (e.g., sensors, medical devices) and the physical environment (e.g., human body). Figure 1.8 illustrates a hierarchical view of the modeling constructs proposed by [Ban12].

Likewise, various approaches have been devised to take into account the entangled nature of distributed and decentralized systems. Some works [Mau+06; Wan+13; Rau+17] considered the direct link between energy/computational constraints with the performances of the activity recognition models. Authors in [Wan+13] investigated the trade-offs between classification accuracy and energy efficiency by comparing on- and off-node schemes. An empirical energy model

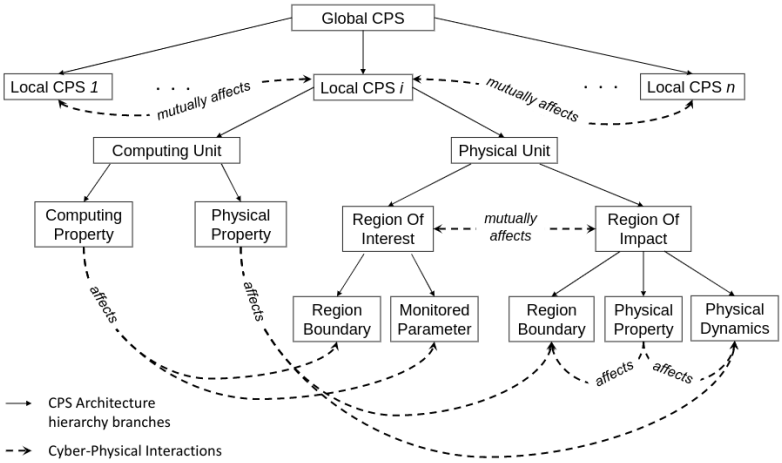


Figure 1.8: Hierarchical view of the modeling constructs proposed by [Ban12] to specify cyber-physical systems.

was presented and used to evaluate the energy efficiency of both systems, and a practical case study (monitoring the physical activities of office workers) was developed to evaluate the effect on classification accuracy. The results show that 40% energy saving can be obtained with a limited 13% reduction in classification accuracy. Similarly, with the goal of analyzing the trade-off between recognition accuracy and computational complexity, authors in [Mau+06] investigated the impact of different sampling rates and other parameters on the performance of activity recognition models. For example, in [OM13], the authors presented a temperature-sensitive routing protocol in wireless body sensor networks for which temperature and heat production are fundamental. These routing protocols take the temperature of the node as a metric in the decision of the routing path. The purpose is to keep the temperature of the node below the safe level and slow down the rate of temperature rise so that it does not harm the human body [OM13].

Although these trade-offs have a direct impact on the learning phase, they are often solely considered at the specific level where they arise. This makes it necessary to propagate these trade-offs, linked to hardware and application aspects, to the level of learning processes.

Relativity of viewpoints

The collective dimension of sensors is important in many situations in order to monitor phenomena of interest. Sensors distributed in various positions of the space provide rich perspectives and contribute in different ways to the learning process. The heterogeneity brought by these configurations in term of views is beneficial but require to be explicitly handled. Indeed, how to reconcile these

different points of view, each having its own perspective which can potentially be redundant or even seemingly contradictory with those of other nodes in the system? How to leverage the relativity between these different perspectives and their relativity with the considered phenomena?

The spatial structure (or disposition) of the sensors deployment and the induced views, the phenomena being monitored are accentuated by the sensors' capabilities, and the perspectives (views) through which the data is collected (position in space, position on the body, video capture modalities, acceleration, gravity, etc.) [AC09; WKA10; HO20]. Moreover, the incomplete and redundant perspectives can lead to confusion of the concepts between them and reduce the performance of the learning independently of the algorithm used. In the case of human activity monitoring from on-body sensor deployments, the sensing devices are generally placed on the following body positions: *waist*, *thigh*, *necklace*, *wrist*, *chest*, *hip*, *lower back*, *trunk*, *shanks*, *ankle*, *pocket*, *hand*, *back pack*, *torso*, *ear*, etc.

A long line of research has focused on the problem of optimal placement and combination of sensors on the body to achieve satisfactory levels of recognition, and many reviews report on this, such as [Ata+11; Att+15]. As an example, Gjoreski et al. [GG11] studied the optimal location of accelerometers among the waist, chest, thigh, and ankle for posture recognition and fall detection. The authors found that a number of sensor configurations are sufficient to correctly recognize most of the postures and fall events. More generally, as reviewed in [Att+15], several works (e.g., [Kar+06; Mat+04; Par+06; YWC08]) provided empirical evidence of the substantial improvements obtained using an accelerometer placed on the waist for the recognition of many activities such as sitting, standing, walking, lying in various positions, running, stairs ascent and descent, vacuuming and scrubbing.

This leads to the research questions: "How to efficiently fuse the relative views provided by the sensing environments? How to efficiently consider the relativity (precisely, the relative positions) of views provided by the sensing environments? How to isolate and temper the exact impact that the sensor's position bias has on the learning process?" It is necessary to meta-model how the views (and, more generally, decentralized learners) interact with each other and incorporate this into the learning process. Therefore, another question that we pursue is: can prior knowledge, e.g., a topological coverage model (see Figure 1.9), be exploited to guide the learning process?

Ethical, privacy, and regulatory constraints

Various regulatory constraints can be found, for example, in the particular case of body area networks and health applications based on on-body sensors deployments, e.g., constraints limiting the exposure to low-frequency fields, circumscribing the risks and potential performance issues that might be associated with wireless co-

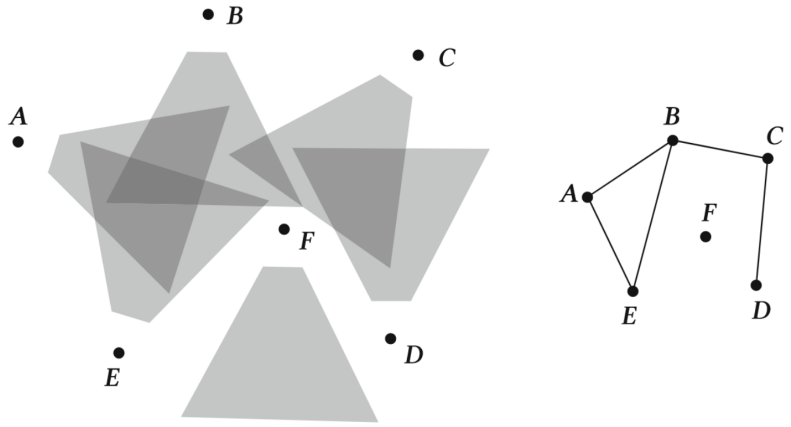


Figure 1.9: Example of prior knowledge about the sensing environment that can be leveraged to improve conciliation of relative views and decentralized learners. From [MC13]: A topological coverage model in the case of a camera network. The model is represented as a graph, where the vertices correspond to the cameras, and the edges indicate pairwise overlap between the cameras.

existence in a shared wireless environment, and ensuring the wireless quality of service. In these kinds of applications, distributed sensing devices often have limited size and computational capabilities imposed by regulatory constraints. For example, IEEE standard 802.15.6 [Sal+16], European Union medical device regulation requirements [Eur17], and ISO 13485:2016 [ISO+16]. A comprehensive review on the regulatory requirements in the specific case of wearable sensors in preclinical and clinical testing can be found in [Rav+19b]. Furthermore, these constraints around physical aspects require a certain adaptation in the processing and transmission paradigms that are used in these devices. Indeed, limited capabilities mean that the sensing devices need to constantly centralize the generated data via short-range transmissions into some kind of gateway. These transmissions, in turn, involve a number of constraints related, this time, to the health issues that they would produce if performed in profusion. In this sense, various standards like [Sal+16] have been set specifically to constrain the transmissions in terms of, e.g., transmission power, latency, packet error rate, network density, body-specific absorption rate, and interference.

A long line of research exists around these types of constraints, and various approaches have been proposed to optimize transmissions. For example, in [OM13], the authors presented an investigation of temperature-sensitive routing protocols in wireless body sensor networks for which temperature and heat production are fundamental. These routing protocols take the temperature of the node as a metric

in the decision of the routing path. The purpose is to keep the temperature of the node below the safe level and slow down the rate of temperature rise so that it does not harm the human body [OM13].

Although providing some substantial improvement in the specific case of transmissions, the particularity of these approaches is that they are ad-hoc and circumscribed to the particular level of transmission. Noticeably, the other levels of processing, e.g., learning processes, are not considered explicitly with regard to these issues. In other words, the different processing levels are compartmentalized. So how to learn while still complying and accounting for the aforementioned legal requirements and regulatory constraints?

1.2.1 Research questions

Following the previously reviewed domain-specific regulation and constraints, which allowed us to further refine the thesis scope, we end up with the following challenging research questions that we aim to tackle so as to build the unifying and integrated framework described earlier.

Research questions: integrating domain knowledge via inputs' structural constraints

- How do characteristics of sensing devices impact the learning process (heterogeneity, distribution skew, etc.)? How to account for the characteristics of the sensing devices and their evolution during deployment in real-life settings?

- How can learners adapt rapidly to the evolution of sensing environments and to the dynamical factors impacting them? Are there representative learning examples that sustain the evolution of the learned models in terms of their decision boundaries?

- How to conciliate different perspectives that exhibit redundancy, contradictory objectives, etc.? How to follow the evolution of the environment, the context, the reglementary constraints, etc.? How to handle the fact that the learned models are often tied to the physical phenomenon it monitors?

- How to account for the heavy cost of performing sensing operations? how to account for the constraints imposed on transmitting data over networks (e.g., energy and computational constraints, limited bandwidth, path loss, etc.)?

- How representative are the end results of sensing operations w.r.t. the example space?

- How do energy and computational constraints shape data acquisition and transmissions, which ultimately impact data quality as well as data availability (the problem of partial or incomplete views)?

Research questions: structuring the learning process guided by the concepts to learn

- How to account for the entanglement between the cyber and physical domains and its impact on the sensing process?
- How to organize computations (learning problems and sub-problems) in order to account for the energy and computational constraints of the deployment-end computing devices?
- How does the semantics of the label space can help to organize these computations?

Research questions: abstracting the context and modeling relativity

- How to efficiently fuse the relative views provided by the sensing environments? how to do so while taking into account legal requirements and regulatory constraints, e.g., privacy-preservation issues and health standards?
- How to efficiently consider the relativity (precisely, the relative positions) of views provided by the sensing environments?
- How to isolate and temper the exact impact that the sensor's position bias has on the learning process?
- How to leverage domain knowledge, e.g., about the structure of the sensing environment, in order to improve the way we isolate and temper the impact of the sensor's position bias?

1.2.2 Aim and Scope

After refining the research questions, we can specify further the concrete objectives that we pursue in order to build the unifying and integrated framework we are aiming at. To this aim, we are set in this research to develop the core building blocks of the proposed unifying framework along with its corresponding methods (or modules), which can efficiently deal with the various issues stemming from distributed sensing environments, like the heterogeneity of data sources, the dynamicity of the deployments, and the heavy cost of sensing and transmitting data. Specifically, this research is divided into four objectives. (1) The development of methods that are robust to the heterogeneity of the end devices in terms of their

sensing characteristics, physical constraints, and the particular contexts they are surrounded by; (2) The development of methods that are able to adapt rapidly to the dynamicity of the deployments and are data-efficient; (3) The development of learning methods that can take into consideration the computing constraints by organizing the learning process in terms of difficulty and can minimize the transmissions; (4) The development of methods that are able to conciliate efficiently between different relative views or different decentralized learners.

1.3 Meta-learning guided by domain knowledge

This thesis takes place in the context of the generalization and pervasiveness of sensing, actuation, and computing capabilities in an effort to enhance the learning process by equipping it with a structural and integrated dimension based on domain knowledge.

In the traditional learning paradigm, learning models typically take the following general form: the learner is supplied with a hypothesis space and training data drawn independently according to some underlying distribution. Based on the information contained in the training data, the learner's goal is to select a hypothesis from the hypothesis space which minimizes some measure of expected loss with respect to the underlying distribution. In such models, the learner's bias corresponds to the choice of the hypothesis space. This is basically how the learning process is viewed from the standpoint of the probably approximately correct (PAC) model, which allows the analysis of the conditions under which learning can be successfully achieved [VC82; Val84; Blu+89]. In this model, the choice of inductive biases is among the most important components.

The inductive biases form the ground upon which the learner can choose one hypothesis that explains the examples it sees. Indeed, the choice of the problem representation or deciding that the hypotheses space takes the form of a class of linear functions or neural networks are a form of bias. The selection of an appropriate set of features to represent the inputs is also in itself a bias. In a sense, the biases guide the learner in electing one hypothesis at the expense of another. Indeed, although difficult, finding the right learning bias makes the actual learning process straightforward. This, however, supposes that the biases are fixed in advance and for the entire course of the model deployment, precluding any form of flexibility and adaptation.

Existing approaches trying to account for the constraints that emerge in the context of the generalization of sensing and actuation capabilities are circumscribed to the processing levels where they arise and often involve the traditional bias-fixing paradigm. The traditional bias-fixing paradigm is not appropriate in these situations. Rather, the learning process has to be equipped with suitable

mechanisms allowing it to be flexible enough and able to accompany the evolution dynamics. As we mentioned previously, this requires us to develop new strategies and, ultimately, a unifying (or integrated) framework for leveraging these domain-related constraints, the existing approaches which have been devised for these specific constraints, and the interplay between all these aspects. There is a need to (i) explicitly consider these aspects within the learning process and ultimately (ii) optimize them in a joint manner and account for their evolution dynamics.

Learning-to-learn models [Sch87; BBC91; Fin18] offer flexibility and a promising level of generalization to overcome the specificities of the real world. To this aim, we will take a meta-modeling approach where we leverage various forms of structures originating from the domain in order to guide the learning process and place the learner in appropriate conditions to carry the learning process. We adopt in this thesis the metalearning paradigm and its ability to learn appropriate inductive biases. Unlike classical models that fix these biases a priori, learning-to-learn models offer promising flexibility and a level of generalization to overcome the specificities of the real world. In particular, we investigate how domain knowledge can be leveraged in order to guide the learning process. For example, the topology of sensor deployments, laws of physics, analytical models describing the phenomena of interest, and dependencies between the concepts to learn can be explicitly incorporated into the learning process in a way that reduces the search space or makes it affordable. The idea is that neglecting prior knowledge about the learning problem at hand unnecessarily makes the learning problem significantly harder.

Naturally, the proposed framework encompasses two complementary pillars: on the one hand, finding appropriate representations that can capture domain knowledge and, on the other hand, exhibiting the corresponding learning mechanisms in the learning processes on which one can act. Let's describe, in more detail, these two pillars.

Meta-learning

Meta-learning can be viewed as a means of reasoning about the learning process and acting on it by providing better inductive biases. Reasoning involves observing how the learning process behaves on different related (or similar) learning problems and how the learning problems are related to each other. More appropriate inductive biases are then devised so as to guide the learner toward certain solutions by further reducing the size of the hypothesis space, adapting the hypothesis space, or providing an ordering on the exploration of the hypotheses. For example, contemporary meta-learning approaches, particularly gradient-based ones, try to act on the learning process by choosing a more efficient initialization [FAL17], generating more efficient optimizers [Bel+17], generating model descriptions [ZL16; SSZ17], choosing an appropriate loss function and evaluation strategy [Sun+18;

KSS19; Bec+21], tuning the learning rate [KBT19], or devising a better metric space [Vin+16]. Ultimately, the goal of meta-learning includes improving the speed of learning and convergence rates, leading the learner to better solutions in terms of performance and robustness, and also equipping the models with explainability with the emergence of higher-level human-interpretable features. Basically, learning to learn boils down to answering the question as Vilalta and Drissi put it: "how can we exploit knowledge about learning (i.e., meta-knowledge) to improve the performance of learning algorithms?".

Often, meta-learning approaches try to reason about their own learning process (or knowledge about learning) by leveraging past experience and exhibiting hyperparameters that can be optimized in order to improve the learning process when facing new tasks or learning configurations. Domain specificities inherent to the generalization of sensing and actuation capabilities bring genuine axes along which meta-learning can be extended, as well as new challenges to further cope with. Meta-learning, as defined, does not fully encompass this notion of taking into account the contextual information that surrounds the data generation process in the context of distributed sensing environments. Existing meta-learning approaches need to be extended following various axes:

- Consideration of the cost of sensing and transmission;
- Consideration of deployment evolution dynamics;
- Consideration of the heterogeneity induced by significant differences across distributed sensing devices in terms of characteristics, physical constraints, and interlacing (or interplay) with cyber-physical elements;
- Organization of calculations related to learning so that this process can be decomposed into several sub-problems that are easier to solve while maximizing the notion of reuse and sharing (transfer) between these sub-problems;
- Allow to reconcile relative perspectives (redundant, missing, seemingly contradictory) or learners carrying out decentralized learning.

On the one hand, there are the base learning processes encompassing the classical learning pipeline and its common building blocks, e.g., pre-processing, segmentation, feature extraction, learning algorithms, and evaluation techniques. This is the lower-level processing layer which is widely studied in the literature and used practically in various applications. On the other hand, the upper-level layer is concerned with the choice of the appropriate learning parameters, the explicitation of the hyperparameters, the definition of the system's boundaries, etc. In sum, at this level of the framework, we are concerned with *what to learn?* and *how to learn it?*. As the base learning setting is widely spread and commonly discussed in the

literature, we will elaborate mainly on the upper-level layer aspects and particularly the meta-learning literature, discuss its key principles, and propose avenues for extending it according to the enlightenments that we get from the massively distributed and decentralized applications reviewed earlier as well as the various domain-related constraints. In Chapter 2, we will review the literature around meta-learning from the early work of [Sch87] to the popular *learning to learn with gradients* framework proposed in [Fin18]. We will expose the different formalisms of meta-learning and introduce an extended set of approaches that ambition to account for the challenges raised by the widespreadness of the sensing and actuation capabilities.

Leveraging domain knowledge

Very often, prior knowledge is available about the learning problems at hand. For instance, images are known to exhibit spatial dependencies, natural language reveals rich structures, and molecules include graph structures. At the risk of unnecessarily making the learning problem particularly harder, learning processes should therefore take into account domain knowledge.

In the context of the generalization of sensing and actuation capabilities, for instance, we are not without the presence of such domain knowledge. Noticeably:

- Availability of equational models that partly describe the phenomena, the topology of deployments, generative models of sensors, models describing the dynamics of deployments, etc. These models would make it possible to reduce the dependence vis-à-vis the real training examples since one can base oneself on these proven models and simply supplement them with parsimonious sampling;

- Availability of semantics in the label space, e.g. atomic (or groups of) labels (or concepts) are related to each other with rich structures. These semantics can be leveraged in order to organize the concepts to learn into different groups according to their proximity (the learning of semantically close concepts is known to be more affordable) then into semantically plausible hierarchies. This would make it possible to exploit what is learned at each level of the hierarchies and facilitate transfer between groups while reducing the quantities of data needed to learn;

- Availability of a priori knowledge describing dependency structures between components of deployments, e.g., geometry of deployments, the way different users perform a given activity (in the case of human activity recognition applications), bio-physical equational models describing the interplay between

different characteristic points of the human body (in the case of monitoring human activities using on-body sensor deployments), etc.

These constitute additional data that have to be integrated into the learning process. These types of data are different in nature from the data manipulated in the classical learning framework and basically describe the phenomena of interest (or problems at hand). They correspond instead to meta-data, which provide contextual information related to the basic data used in the classical learning framework, such as the quality of the data, the importance of the data source used to generate the data, the dependence between various data sources, the dependence across the atomic (or groups of) concepts to learn from the data, etc. These types of data do not have the same status as regular data and, therefore, cannot be directly manipulated in basic learning processes. Instead, these types of data intervene in a higher-level process analogous to the one that is performed in existing meta-learning approaches, where the idea is to reason about the regularities that appear across learning problems in order to improve learning performances on new (never-seen) problems. Here, instead of only relying on mere statistical regularities, domain knowledge constitutes explicit regularities that can be additionally used to guide the learning process.

The problem shifts from simply learning using data to leveraging domain knowledge to put the learner in optimal conditions to carry out learning. The proposed framework comes with a lot of challenging problems that have to be solved, noticeably: Explicitly considering domain specificities within the learning process subsequently poses concrete issues regarding the way we can integrate these aspects into the learning process. How to integrate domain knowledge into the learning process? Indeed, constraints do not all have a direct equivalent in terms of hyperparameters in learning processes. So, how to express (or represent) domain knowledge in a way that it can be easily integrated into the learning process?

This could be related in some ways to the path being undertaken to bridge the gap between knowledge representation and reasoning approaches, on the one hand, and machine learning, on the other hand [Mit+18; DKT07; De +19]. Indeed, it is more appropriate to see it as the different ways a priori knowledge (or the representation of it) is "translated operationally" into the deployed models and which hyperparameters could be exhibited to ease this process. In this matter, one question that remains is, in the concrete case of neural networks, for example, the link (or relation) between prior knowledge and the weights or the structure of the neural networks. Integrating prior knowledge, in a principled fashion, into deployed models also remains an open question, as is the case for the other aspects we are investigating. Additionally, this correspondence has the advantage of paving the way for interpretability and explainability both for the traditional goal of complying with legal constraints and also for the goal of mechanizing the

evolution of the base models to remain in compliance when these legal constraints evolve. In other words, provide this correspondence with a more "productive" dimension. These questions are discussed throughout the entire thesis. For example, one approach that is investigated in Chapter 6 consists in using elements from the special Euclidean group to represent the relative geometry of the sensor deployments and simultaneously act on the structure of the neural architectures to enforce such prior knowledge.

Furthermore, considering the various domain-related constraints in a joint manner raises, for its part, challenges related to how to optimize and reason (in relation to the learning process) about the prior knowledge that can be integrated into the learning process. For example, Chapter 5 investigates different approaches to construct hierarchical structures that are suitable for organizing the learning process, i.e., optimizing the structure w.r.t. the learning process to maximize transfer and the learning performances. Also, in Chapter 4, the interplay between different prior models from the domain (physicochemical models of the reactants) and their impact on the learning process are investigated.

This research is organized around three axes. (1) The development of methods that leverage domain knowledge to guide the learning process by selecting appropriate learning examples and augmenting the example space in suitable regions; (2) The development of methods that leverage the semantics of the label space to organize the learning process; and (3) The development of methods that efficiently conciliate different relative views provided by distributed sensing environments, using domain knowledge, e.g., in the form of prior geometrical information about the structure of the deployment. In the next section, we provide a detailed overview of our core contributions in the sense of building the proposed framework.

1.4 Contributions

The thesis is organized around three axes where we describe and evaluate different strategies for structuring the learning process in the context of generalized sensing capabilities. The presentation of these axes is preceded by a background chapter on meta-learning approaches. Implications of the work and its future directions are discussed in a concluding chapter. A summary of the proposed axes and specific contributions are described in the following subsections.

Integrating domain knowledge via structural constraints

In Chapter 4, we will be concerned with the way one can control the internals of the learning processes according to the domain-related constraints. In this chapter,

we propose two novel approaches that leverage domain knowledge to select and augment learning examples.

The main bottlenecks being dealt with in this chapter are the heterogeneity of the data sources and the cost of sensing and transmitting learning examples. We take into account the heterogeneity induced by significant differences across distributed sensing devices in terms of characteristics, physical constraints, and interleaving (or interplay) with cyber-physical elements.

For this, we exploit the availability of equational models, which partly describe the phenomena, the topology of the deployments, the generative models of the sensors, models describing the dynamicity of the deployments, etc. The idea is to reduce the dependence on real training examples since we can build on these proven models and simply supplement them with parsimonious sampling. In particular, we will investigate different strategies for incorporating domain knowledge into the learning processes by imposing structural constraints on the base models, i.e., constraining the inner workings of the base model and its structure. The interplay between different prior models from the domain and their impact on the learning process are investigated. The approaches presented in this chapter are based on the following works [OHB19; HO20; OHB21; HO23].

Structuring the learning process guided by the concepts to learn

In Chapter 5, we will investigate ways of organizing the learning process so that it can be broken down into several sub-problems that are easier to solve while maximizing the notion of reuse and sharing (transfer) between these sub-problems.

We build upon the possibility of exploiting the semantics of the label space in order to organize the concepts to be learned into different groups according to their proximity (the learning of semantically close concepts is more affordable) then their hierarchization which would make it possible to exploit what is learned at each level and facilitate transfer between groups. We will focus on different strategies devised specifically to structure the concepts to learn, i.e., the semantics of the label space.

The other axe that we pursue to guide (or control) the learning process is structuring what to learn in a way that the traversal of the hypothesis spaces becomes fragmented according to the a priori knowledge. The idea of structuring the concepts to learn draws a lot of similarities with the way humans learn new concepts and acquire knowledge. Specifically, we will investigate two different approaches for structuring the learning process. Both approaches are based on optimizing the structure of the atomic concepts to learn first before proceeding to the actual learning. This way, the more general groups of concepts are first

learned together before more specific groups of concepts and, ultimately, the atomic concepts are learned. The approaches presented in this chapter are based on the following works [OHA21a; OHA22].

Abstracting the context and modeling relativity

In Chapter 6, we will focus on the collaborative aspects of the massively distributed sensing nodes and the ways conciliation of decentralized learners can be improved. We investigate approaches that can efficiently fuse the relative views provided by the sensing environments and conciliate the decisions taken by decentralized learners while considering their relativity. Relative perspectives can be redundant, missing, or seemingly contradictory to each other, and naive conciliation in the decentralized learning setting leads to poor learning performances.

To improve this process, we explore how to leverage prior knowledge about the sensing deployments. Noticeably, leveraging topological models describing the disposition of the sensing devices and equational models describing the phenomena considered for learning. For example, in the case of HAR applications considered in the empirical evaluations of the chapter, we exploit the geometrical information about the on-body disposition of sensors as well as bio-physical models describing the spatio-temporal dynamics of the movements, i.e., how the body parts interact with each other while the activities are performed.

These additional prior models are expressed in appropriate mathematical operators, which are further used to constrain the architectures of the neural networks. This ultimately acts on the hypothesis space that the learner has to explore by reducing the admissible set of hypotheses to only those satisfying the constraints expressed above. The approaches presented in this chapter are based on the following works [OH22; HO22].

1.5 Impact and applications

Throughout this thesis, we will approach the problem of learning in the context of distributed and decentralized data using concrete IoT applications, namely, infant cries recognition, turbocompressor vibration monitoring, synthesis of new materials in the industry, and human activity recognition from wearable sensor deployments. In the following, we provide a concise description of each of these applications and the problems being tackled.

Infant cries recognition This application is concerned with the monitoring of infant comfort using an IoT-based solution encompassing a set of distributed

and decentralized elements, including a voice activity detector, environment monitoring module, and soothing module [OHC17b]. The goal is to train the main module to learn infant cries signatures and their corresponding comfort situation (e.g., hungry, need attention, or afraid). The main module processes the infant's vocalizations, and the additional modules provide contextual information to the main module, which uses them to guide the learning process. A number of validated signatures of pre-cry episodes are provided by pediatricians and domain experts [Dun09; Coo+12]. Additionally, deployed devices, their functioning, and their transmissions are strictly regulated by health constraints and domain-specific standards. The idea is to maintain a required level of exact predictions while prohibiting a high burden on the deployed devices. We are witnessing increasingly higher demands for health monitoring, ambient intelligence, and assisted living applications, which is due, for example, to the aging population and societal evolutions. Providing answers to the aforementioned challenges is key to the wider adoption of such applications, and the integrated perspective we undertake toward these challenges is promising.

Turbocompressor vibration monitoring In this application, we explore the problem of condition monitoring of large industrial equipment from deployments of vibration sensors. Different approaches are used for monitoring these kinds of industrial equipment and can be categorized into model-based, data-driven, and experience-based approaches [Tob+12]. The applicability of these approaches is usually assessed based on three criteria including cost, precision, and complexity. This application involves a set of expert-defined domain specifications such as deployment specification, which describes the disposition of vibration sensors and the way they are mounted on specific parts of the industrial equipment, and domain standards, which define the different thresholds that the system must remain within. This is probably one of the areas that witness large efforts toward the integration of domain knowledge into the learning processes. The integrated meta-modeling approach pursued in this thesis provides some concrete answers to the inherent challenges encountered in this area.

Synthesis of new materials In this application, we investigate the problem of material design in the industry, where the core function is to accelerate the synthesis of new materials with good properties [Ayk+19; Sev+19; Tab+18]. We focus in particular on the distributed and decentralized scenario featured by these kinds of applications where concrete data (corresponding to real experiments) are distributed over the experimental state space and scarce (due to the costly efforts required to produce these examples). The goal is to build approximation models for the entire state space from these distributed partial views. The study of this

application sees the opportunity to leverage domain analytical models describing the kinetics and thermodynamics of the chemical phenomenon involved.

Human activity recognition In this application, we study the classification of human activities like running or biking using data generated from on-body sensor deployments. Learning in this context imposes to conciliate the views provided by each type of sensing modality (like accelerometers and gyroscopes, which provide different information) and the respective on-body location of each sensor (the location on its own and its relative position w.r.t. other sensors and w.r.t. on-body referential). Activity recognition is addressed traditionally according to the following predefined chain [BBS14]: the labeled examples generated from the sensors are (i) segmented into short sequences; which are (ii) pre-processed; and (iii) from which discriminative features are extracted; (iv) before being fed into a machine learning algorithm responsible of finding the mapping towards the activities. This is the HAR pipeline that we investigated from the perspective of the integrated framework in one of our previous works [OH19; HO21b]. We noticeably exhibited a set of hyperparameters from the HAR pipeline as well as the corresponding domain aspects, which are then optimized and assessed in a systematic manner. Overall, this application brings into play many different aspects of the domain, such as the notion of deployment topology, biomechanical models of the movements, and the relative importance of the sensing modalities. The broader impact of HAR models is a key enabler for the development of more effective assisted living and ambient intelligence applications. The integrated perspective for which we are laying down the foundations in this thesis is appropriate for dealing with the inherent challenges facing HAR.

1.6 Publications

The work presented in this thesis is based on the following publications.

1. Aomar Osmani, Massinissa Hamidi, and Pegah Alizadeh. "Clustering approach to solve hierarchical classification problem complexity". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 36. 7. 2022, pp. 7904–7912

2. Massinissa Hamidi and Aomar Osmani. "Context Abstraction to Improve Decentralized Machine Learning in Structured Sensing Environments". In: *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*. Springer. 2022

3. Aomar Osmani and Massinissa Hamidi. “Reduction of the Position Bias via Multi-level Learning for Activity Recognition”. In: *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer. 2022, pp. 289–302

4. Aomar Osmani, Massinissa Hamidi, and Salah Bouhouche. “Augmented Experiment in Material Engineering Using Machine Learning”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 10. 2021, pp. 9251–9258

5. Aomar Osmani, Massinissa Hamidi, and Pegah Alizadeh. “Hierarchical Learning of Dependent Concepts for Human Activity Recognition”. In: *PAKDD*. Springer. 2021

6. Massinissa Hamidi and Aomar Osmani. “Data Generation Process Modeling for Activity Recognition”. In: *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*. Springer. 2020

7. Massinissa Hamidi, Aomar Osmani, and Pegah Alizadeh. “A Multi-View Architecture for the SHL Challenge”. In: UbiComp-ISWC ’20. New York, NY, USA: Association for Computing Machinery, 2020, pp. 317–322

8. Aomar Osmani, Massinissa Hamidi, and Salah Bouhouche. “Monitoring of a Dynamical System Based on Autoencoders”. In: *proceedings of the 28th International Joint Conference on Artificial Intelligence, IJCAI*. 2019, pp. 1836–1843

9. Aomar Osmani and Massinissa Hamidi. “Hybrid and convolutional neural networks for locomotion recognition”. In: *Proceedings of the 2018 ACM UbiComp/ISWC 2018 Adjunct, Singapore, October 08-12, 2018*. ACM. 2018, pp. 1531–1540

10. Aomar Osmani, Massinissa Hamidi, and Abdelghani Chibani. “Platform for Assessment and Monitoring of Infant Comfort”. In: *2017 AAAI Fall Symposia, Arlington, Virginia, USA, November 9-11, 2017*. 2017, pp. 36–44

11. Aomar Osmani, Massinissa Hamidi, and Abdelghani Chibani. “Machine Learning Approach for Infant Cry Interpretation”. In: *Tools with Artificial Intelligence (ICTAI), 2017 IEEE 29th International Conference on*. IEEE. 2017, pp. 182–186

12. Massinissa Hamidi and Aomar Osmani. “Human Activity Recognition: A Dynamic Inductive Bias Selection Perspective”. In: *Sensors* 21.21 (2021), p. 7278

13. Massinissa Hamidi and Aomar Osmani. “Domain models for data sources integration in HAR”. in: *Neurocomputing* 444 (2021), pp. 244–259

Chapter 2

Preliminaries, Notations, and Meta-learning Models

This chapter discusses learning-to-learn (or meta-learning), its key components, and the recently proposed lines of predominant approaches inspired by gradient-based meta-learning. Besides introducing the key concepts of meta-learning, throughout this chapter, we contextualize our core contributions and provide pointers to the corresponding chapters.

Broadly speaking, as framed by Mitchell et al., "a computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E". To make a machine solve a learning problem, one must therefore identify the class of tasks, the measure of performance to be improved, and the source of experience [Mit+97]. Operationally, a learning process is then defined to guide the learner towards a solution to the learning problem. The idea of meta-learning, in a nutshell, is to automate (or abstract away) the design choices of the learning process so as to continuously improve upon experience and guide the learner to attain better solutions more rapidly. In the following, we will provide essential background about meta-learning. Whereas we refer the reader to [Mit+97; Bis06], and many other excellent resources for a more detailed presentation of machine learning aspects. Note that we refer in the following indiscriminately to learning-to-learn and to meta-learning and that we concentrate on the risk minimization framework as a formal model for machine learning.

The rest of this chapter is organized as follows. After reviewing the basics of the learning process (§ 2.1), we will dig into the learning-to-learn (or meta-learning) paradigm (§ 2.2): we will take a look at the way it is defined in the literature, how

is it formalized, and what concrete implementations were proposed alongside the way it relates to other closely related domains. In Section 2.3, we will describe one of the prominent approaches of learning-to-learn, gradient-based meta-learning (GBML). This type of approach, which relies on the notion of gradient, was popularized recently by the model-agnostic meta-learning (MAML) algorithm proposed by Finn, Abbeel, and Levine. The remaining sections concentrate on the key components of the learning-to-learn paradigm. In Section 2.4, we discuss the literature around the emergence of a fundamental distinction between domain-agnostic and domain-specific parameters in the learning-to-learn models. Links between domain expert knowledge and these elements are established, which will be further investigated in the following chapters. Furthermore, data and meta-data are two important components that determine how training at a meta-level is performed. We will evoke how domain expert knowledge is incorporated into the learning process via the meta-level training phase. These are discussed in Section 2.5. We elaborate in Section 2.6 on an important aspect of learning-to-learn, which is task-relatedness. The task-relatedness aspects allow us to make the connection with federated learning via the various structures that emerge in the distributed and decentralized applications at different levels.

The introduction to the federated learning setting is deferred to Chapter 3, where we focus on the induced heterogeneity across clients and objective inconsistency. We also review how these phenomena are dealt with in the literature. These constitute the key challenges that we propose to solve in this thesis. For an in-depth introduction to the federated learning field, we refer the reader to the excellent recent entries [Kai+19] and [Wan+21].

Of course, this chapter does not substitute itself to the excellent book "*learning to learn*" [TP98] by Thrun and Pratt as an introduction to the fundamental principles of meta-learning. Also, note the excellent review of the recent state-of-the-art in [Hos+21], which presents meta-learning and related fields from an interesting taxonomy.

2.1 Learning process basics

According to the PAC model of machine learning and its variants [VC82; Val84; Blu+89], supervised learning models typically take the following general form: the learner is supplied with a hypothesis space $\mathcal{H}$ and training data $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), ...\}$ drawn independently according to some underlying distribution P on $X \times Y$. Based on the information contained in the training data, the learner's goal is to select a hypothesis $h : X \to Y$ from $\mathcal{H}$ minimizing some measure $er_P(h)$ of expected loss

with respect to P, i.e., $er_P := \mathbb{E}_{(\mathbf{x},y)\sim P}\ell(h(\mathbf{x}), y)$ [1]. In such models, the learner's bias is represented by the choice of $\mathcal{H}$; if $\mathcal{H}$ does not contain a good solution to the problem, then, regardless of how much data the learner receives, it cannot learn [Bax00]. In general, models of supervised learning include the following: an input space X and an output space Y, a probability distribution P on $X \times Y$, as well as a loss function $\ell : Y \times Y \to \mathbb{R}$. One also has to define a hypothesis space $\mathcal{H}$, i.e., how to represent the hypotheses (or functions). For example, the hypotheses can be represented as a space of linear functions $h_\theta(\mathbf{x}) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots$, mapping inputs $\mathbf{x} = (x_1, x_2, \ldots)$. Notice that the hypotheses are now subscripted with $\theta = \{\theta_0, \theta_1, \theta_2, \ldots\}$ which are basically the weights parameterizing the space of linear functions mapping X to Y.

For example, in the case of human activity recognition, one possible mapping is that X would be the set of observations generated by the on-body sensor nodes, Y would be the set of target activities (walk, run, etc.), and the distribution P would be peaked over different episodes during which users perform one of the target activities. The learner's hypothesis space $\mathcal{H}$ would be a class of neural networks mapping the input space X to Y. This is a classification problem, and the loss, in this case, would be as simple as a discrete loss $\ell(h(x), y) = \mathbb{1}_{[h(\mathbf{x})\neq y]}$, where $\mathbb{1}_{[\pi]}$ equals 1 if the predicate π is true and 0 otherwise. This is called the $0/1$ loss and looks for accuracy, i.e., how many times the learner got the right predictions. In the case of multi-class classification, this loss does not care about how the errors are made. Besides explicitly considering how the errors are made (in the categorical form of the loss function), cross-entropy can be thought of as a relaxation of the $0/1$ loss and is a convex function (as opposed to the $0/1$ loss for which authors, e.g., in [NS13], provide direct optimization methods). In the regression framework, ℓ can be some measure of distance between the prediction and the target, e.g., squared loss $\|\cdot\|^2 = \frac{1}{2}\big(h(\mathbf{x}) - y\big)^2$.

The biases constitute the ground upon which the learner can choose one hypothesis that explains the examples it sees. Indeed, the choice of the problem representation or deciding that the hypotheses space takes the form of a class of linear functions or neural networks are a form of bias. The selection of an appropriate set of features to represent the inputs is in itself a bias. In a sense, the biases guide the learner in electing one hypothesis at the expense of another. Indeed, although difficult, finding the right learning bias makes the actual learning process straightforward. Therefore, two important features of bias are strength (reduction factor of hypothesis space) and correctness [Utg86a]. Besides the fact that bias selection generally involves the inputs of domain experts via its design

[1] Alternatively to empirical risk minimization, maximum likelihood estimation tries to solve a probability density estimation problem, where the goal is to approximate, for example, the unknown density $P(y|x)$ given observed data.

choices, both features of bias are subject to trade-offs, making picking the right inductive bias often one of the hardest problems in machine learning [Bax00].

In addition to the definition of the space of hypothesis and the learning examples, the learner is supplied with an algorithm that searches for the optimal hypothesis. For example, when the hypotheses are parameterized and the errors are differentiable w.r.t. these parameterizations, gradient descent can be used to search through the hypothesis space. This is the case with neural networks. According to the choice made for the components of the learning process, the resulting optimization landscape varies in terms of their properties to greater extents. Indeed, apart from differentiability aspects, the optimization landscape can be convex or non-convex [BV04]. In this thesis, we mainly use gradient-based learners as it has been proven to be practically efficient in a wide variety of contexts. Although efficient, these kinds of learners define complex non-convex optimization landscapes which are highly challenging to explore, as the efficient optimization algorithms designed for convex optimization are discarded. Besides being non-convex, the complexity of the optimization landscape is characterized by many challenging properties which have been the center of attention of the community [SMG13; FB17; Fle21]. For example, it was observed in the optimization landscape the presence of low curvature regions and many local minima, which have been found to be relatively low cost compared to the global minima [SMG13] or the fact that all local minima are global [LB18]. These models also suffer from high instability where ϵ-small perturbations can completely modify the optimization landscape [Sze+15]. Inspired by these observations and the properties of the optimization landscape, first-order methods like gradient descent and their derivatives have been proposed to tackle the exploration of such landscapes. Gradient descent starts with a random initialization, $\theta^{(0)}$, of the learner's parameters which is then continuously updated during the learning process as

$$\theta^{(t)} = \theta^{(t-1)} - \eta^{(t-1)} \nabla \ell\big(\theta^{(t-1)}, B^{(t-1)}\big) \tag{2.1}$$

where $\{\eta^{(t)}\}_{t\in\mathbb{N}}$ is a sequence of learning rates and $\nabla\ell$ is the gradient w.r.t. θ evaluated at $\theta^{(t-1)}$ with (a subset of) data $B^{(t-1)}$. Here, design choices involving either the learning rates (or their sequencing) or the way data are presented to the learner (batch, mini-batch, using a particular sequencing, etc.) have major impacts on the learning process. More broadly, inductive biases are the core components of any machine learning system and, as such, can be similarly subject to learning, i.e., choosing them appropriately can lead to better solutions, rapidly.

2.2 Learning to learn: improve with experience

Learning to learn (or meta-learning) can be viewed as a means of reasoning about the learning process and acting on it by providing better inductive biases. Reasoning involves observing how the learning process behaves on different related (or similar) learning problems and how the learning problems are related to each other. More appropriate inductive biases are then devised so as to guide the learner toward certain solutions by further reducing the size of the hypothesis space, adapting the hypothesis space, or providing an ordering on the exploration of the hypotheses. In this sense, early systems, including Shift to a Better Bias (STABB) [Utg86b] or Variable Bias Management System (VBMS) [RST87], tried to look for dynamic forms of bias, while contemporary approaches, particularly gradient-based ones, try to act on the learning process by choosing a more efficient initialization [FAL17], generating more efficient optimizers [Bel+17], generating model descriptions [ZL16; SSZ17], choosing an appropriate loss function and evaluation strategy [Sun+18; KSS19; Bec+21], tuning the learning rate [KBT19], or devising a better metric space [Vin+16] (see illustrative approaches bellow for more details). Ultimately, the goal of meta-learning includes improving the speed of learning and convergence rates, leading the learner to better solutions in terms of performance and robustness, and also equipping the models with explainability with the emergence of higher-level human-interpretable features. Basically, learning to learn boils down to answering the question as Vilalta and Drissi frame it in [VD02]: "how can we exploit knowledge about learning (i.e., meta-knowledge) to improve the performance of learning algorithms?".

Predominantly, learning to learn (or meta-learning) is tightly linked to human learning and how humans acquire knowledge from a continual stream of tasks [TP98]. It is suggested by Lake et al. as a fundamental component to achieving human-level intelligence, along with compositionality and causality. Indeed, the streams of tasks may encompass highly dissimilar tasks nevertheless, humans are able to make correct inferences that go far beyond what they have encountered. This is possible thanks to the strong inductive biases (or prior knowledge) which are accumulated throughout experiences [Gri+10]. Humans acquire this prior knowledge via "learning-to-learn," i.e., they learn how to generalize [TP98]. More precisely, they learn how to compare and make parallels, how to continually learn, how to evaluate, etc. This knowledge can thus be used flexibly in various ways so as to match new situations or new tasks. Often, learning a particular task can benefit from making parallels with how other related tasks were learned before, which is even more actual as inductive biases are often shared to some extent with other related tasks [Lak+17]. As a result, besides accelerating the way new tasks are learned, humans are able to generalize correctly from fewer examples as well as learn richer representations.

For Thrun and Pratt, given (i) a family of tasks, each of which comes with (ii) a training experience, and (iii) a family of performance measures (e.g., one for each task), meta-learning algorithms are defined as those capable of improving their performance at each task while accumulating experience with the number of tasks. In this definition, the notion of task takes a prominent position as it discards algorithms that do not leverage the presence of other learning tasks. We discussed above about the advantages of learning from streams of tasks from the perspective of how humans learn. The ability of a learner to accumulate strong inductive biases while encountering different tasks is crucial to improving artificial intelligence and has been studied theoretically, e.g., [Bax00; MPR16]. Indeed, while studying approaches for learning data representations from multiple tasks, authors in [MPR16] establish theoretical conditions whereby representation learning is more advantageous in multi-task regimes than in independent task regimes. Convergence and generalization rates are provided, which, besides the sample size and the intrinsic data dimensionality, depend on the number of tasks (see discussion in § 3.6 about the impact of task-relatedness on convergence and generalization rates and in § 2.3 about feature learning vs. feature reuse and how observing multiple tasks helps in finding and refining such features) [2].

With this definition, machine learning sees the introduction of the notion of families of (diverse) tasks that the learner should leverage in order to generalize well and adapt easily to new unseen tasks. Inspired by the formalism in [Sch16], rather than considering a unique set of training data (§ 2.1), we consider a domain D of possible experiences $s \in D$, each having a probability $p(s)$ associated with it. Let T be the available training experience at any given moment. Training experience is a subset of D, i.e., $T \in D_T \subset \mathcal{P}(D)$, where $\mathcal{P}(D)$ is the powerset of D [3]. To highlight the notion of a learner encountering a series of learning tasks one after the other, a learner is referred to as an agent π_θ which is parametrized by $\theta \in \Theta$. A task associates a performance measure [4] $\phi : (\Theta, D) \to \mathbb{R}$ with the

[2] Multi-task learning (MTL) consists in learning several related tasks jointly so as to improve the generalization capabilities of the resulting model on these same tasks during testing. Meta-learning, on the other hand, aims at improving generalization capabilities on totally new tasks that have never been encountered before. Recent efforts towards bridging the gap between multi-task learning and meta-learning have been pursued in [WZL21]. In particular, the authors investigated ways of combining fast adaptation characteristics of meta-learning and the efficient training procedures of the multi-task learning approaches. In this sense, the authors show that from an optimization perspective, multi-task learning and the particular class of gradient-based meta-learning algorithms can be expressed using the same optimization formulation.

[3] As we can notice, this formalism is convenient for streaming applications where data is available incrementally and not in a batch fashion.

[4] Note in this formalism that the performance measure is not fixed but flexible and can be assigned at runtime, which is convenient in continual (or incremental) learning settings such as streaming data.

agent's behavior for each experience. The expected performance, denoted by Φ, of an agent on D corresponds to:

$$\Phi(\theta) = \mathbb{E}_{s \in D}[\phi(\theta, s)] \tag{2.2}$$

A learning algorithm $L : (\theta, D_T) \rightarrow \theta$ is defined as a function that changes the agent's parameters θ based on training experience so that its expected performance Φ increases[5]. More formally, we define the learning algorithm's expected performance gain δ to be:

$$\delta(L) = \mathbb{E}_{\theta \in \Omega, T \in D_T}[\Phi(L(\theta, T)) - \Phi(\theta)] \tag{2.3}$$

Any learning algorithm must satisfy $\delta > 0$ in its domain. That is, it must improve expected performance. Who says tasks also says transfer learning. With this formulation, one may be tempted to set up mechanisms that simply carry out the transfer from one task to another. Indeed, transfer learning proved efficient for few-shot learning, where the idea is to exploit inductive biases leaned on one task so as to perform better on a different task. As Schmidhuber states, while making an analogy with how even simple neural networks exhibit the ability to learn new images faster through pre-training on other images, learning-to-learn is not just transfer learning. The notion of how meta-knowledge (or inductive biases) are meta-learned should be explicitly highlighted and handled with appropriate mechanisms.

The notion de metaknowledge is fundamental in any meta-learning system. It corresponds to particular aspects of the learning algorithm (or process) that can be modified to improve its performance on a given learning problem (or a succession of these). The modified version of the learning algorithm should become better than the original version. This meta-knowledge is gained through experiences and the succession of tasks the learner is confronted with. Hospedales et al. in [Hos+21] categorize meta-knowledge, also referred to as across-task knowledge, as the answer to "what to meta-learn?" While authors in [LBG15] highlight two sources from which a meta-learning system can gain experience: meta-knowledge extracted from previous learning episodes on a single dataset (or task) or from different domains or problems (see Sections 2.5 and 2.4 for more details). Mathematically, meta-knowledge is formalized by Schaul and Schmidhuber in [SS10] by what they refer

[5]Note that the learning algorithm may also be assumed to be an atomic (one-shot) process that is executed until completion going from an initial configuration of the parameters $\theta^{(i)}$ to another configuration $\theta^{(j)}$. Of course, there are intermediary configurations that correspond, for example, to the configuration of parameters at the end of an epoch in the case of neural network training.

to as meta-parameters (or hyperparameters) [6]. These are the learning algorithm's modifiable components, μ. The learning algorithm is now parameterized by $\mu \in M$, i.e., $L_\mu : (\theta, D_T) \rightarrow \theta$ [7]. A meta-learning algorithm, $ML : (M, D_T) \rightarrow M$, is defined to be a function that changes the meta-parameters of a learning algorithm based on training experience so that its expected performance gain δ increases:

$$\mathbb{E}_{\mu \in M, T \in D_T}[\delta\big(L_{ML(\mu,T)}\big) - \delta\big(L_\mu\big)] > 0, \tag{2.4}$$

where $\mu' = ML(\mu, T)$ are the updated meta-parameters. For example, ensemble methods like Bagging [Bre96] and Boosting [Sch90], which proceed by combining the outputs of multiple-base-level classifiers, can be represented in the formalism of Schaul and Schmidhuber as follows: D: input/class samples; D_T: $\mathcal{P}(D)$; ϕ: classification errors; π_θ: set of base-level classifiers; θ: parameters of each classifier; L_μ: supervised learning; μ: number of classifiers, data subsets with sample weights; ML: Boosting.

Besides determining which aspects of the learning algorithm are essential in improving the performance, there is a need to provide appropriate mechanisms to reason about past experiences and run useful modifications on the suitable aspects. Indeed, as Schmidhuber frames it: "true learning-to-learn (L2L)" is not just about learning to adjust a few hyperparameters.

"Radical L2L is about encoding the initial learning algorithm in a universal language (e.g., on an RNN), with primitives that allow to modify the code itself in arbitrary computable fashion. Then surround this self-referential, self-modifying code by a recursive framework that ensures that only "useful" self-modifications are executed or survive"

Although very often used to process time series, recurrent neural networks (RNNs) have been proposed for meta-learning. Indeed, their sequential and recursive fashion of processing inputs makes them biased toward these types of data, which allows them to capture temporal dependencies. The recursive processing of RNNs

[6] Note that meta-learning and hyperparameter optimization approaches, e.g., Bayesian optimization, basically boil down to solving a nested optimization problem. However, they differ in terms of the experimental settings in which they are evaluated [Fra+18]. Often in hyperparameter optimization approaches, data is sampled from a single task, while in meta-learning, we deal with a succession of tasks.

[7] It is more convenient to think about the meta-parameters of the learning algorithms as those controlling the dynamics of the learning process, such as the learning rate or weight decay. In this case, along with the actual model's parameters, we will have the definition of the model's architecture, such as the number of layers, number of neurons per layer, or type of activation function. However, this leads to discarding some aspects from being considered as hyperparameters such as the architecture of the learning model (i.e., the parameters θ).

allows them to learn how to update their own weights. Indeed, they can be expressed in the above formalism as follows: D: input-target samples; D_T: $\mathcal{P}(D)$; ϕ: MSE; π_θ: RNN; θ: network activations; L_μ: RNN; μ: network weights; ML: back-propagation through time. This is one reason they are often used as controllers in different meta-learning approaches, as we will see in the following. Note that various improvements have been brought to RNNs since they appeared, noticeably to deal with one of their major drawbacks, i.e., the poor performances when confronted with very large time lags between significant events. Improvements to these models include, for example, Long short-term memory (LSTM) networks [HS97] and gated recurrent units (GRU) [Cho+14]. Besides displaying self-referential and self-modifying capabilities, RNNs and their related models have been leveraged to build a long line of meta-learning approaches ranging from meta-learning the learning rate and the initialization of the learner's weights to the description of the learner's model (or architecture).

Recurrent networks are used to learn more appropriate learning rates and initialization of the learner's weights. For instance, a long line of approaches was constructed around recurrent neural networks (and their derived models such as LSTMs [HS97] and transformers [Vas+17]) as meta-level, which are trained to predict the control parameters of the learners at the base level. Figure 2.1 illustrates this principle exemplified on the approach proposed by Ravi and Larochelle. Indeed, in this work [RL16], authors train a meta-learner LSTM to learn an up-

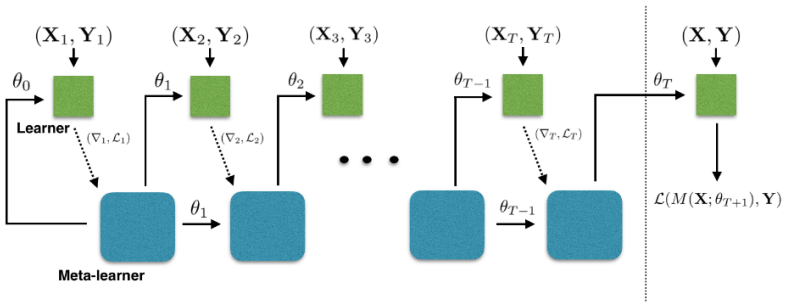


Figure 2.1: Computation graph of the LSTM controller highlighting how it provides the base learner with appropriate learning rates and initialization of the base learner's weights. Figure from [RL16].

date rule for training a base-level neural network. They leverage the observation that the update of the gradient descent, used to train deep neural networks, i.e., $\theta^{(t)} = \theta^{(t-1)} - \eta^{(t)} \nabla \ell(\theta^{(t-1)})$ (see Eq. 2.1), resembles the update of the cell state in an LSTM, i.e. $c_t = f_t \odot c_{t-1} + i_t \odot \hat{c}_t$. Concretely, the correspondence is made by setting the parameters of the LSTM cell, i.e., f, c, i and $\hat{c}$, as follows: $f_t = 1$, $c_{t-1} = \theta^{(t-1)}$, $i_t = \eta^{(t)}$, and $\hat{c}_t = -\nabla \ell(\theta^{(t-1)})$. With this, authors set the hyper-parameters of the base-level learner to be the cell state of the meta-level LSTM

model, i.e., $c_t = \theta^{(t)}$, and the candidate cell state as $\hat{c}_t = \nabla\ell\big(\theta^{(t-1)}\big)$. meaning that the learning rate is a function of the current parameter value $\theta^{(t-1)}$, the current gradient $\nabla\ell(\theta^{(t-1)})$, the current loss ℓ, and the previous learning rate i_{t-1}. With this information, the meta-learner should be able to finely control the learning rate so as to train the learner quickly while avoiding divergence. Furthermore, by leveraging other gates of the LSTM cell the learner can escape from bad local optima by shrinking its parameters. Indeed, if the learner struggles to escape from a bad local optimum, i.e., the loss is high, but the gradient is close to zero, forgetting parts of its previous weights would allow it to explore other parts of the optimization landscape. This can be done using the LSTM's forget gate.

Recurrent networks are also used to generate the computation graph of optimizers used to train neural networks. A growing number of more complex optimizers other than the well-known SGD were proposed in the deep learning literature, each of which tries to cope efficiently with the optimization landscape generated by such models, e.g., RMSProp [HSS12] which exploits the magnitudes of recent gradients of a given weight to correct the learning rate [8], or Adam [KB14]. The idea is to meta-learn these optimizers in the same fashion as the learning rate, or the base level learner's initialization is meta-learned. Authors in [Bel+17] pro-

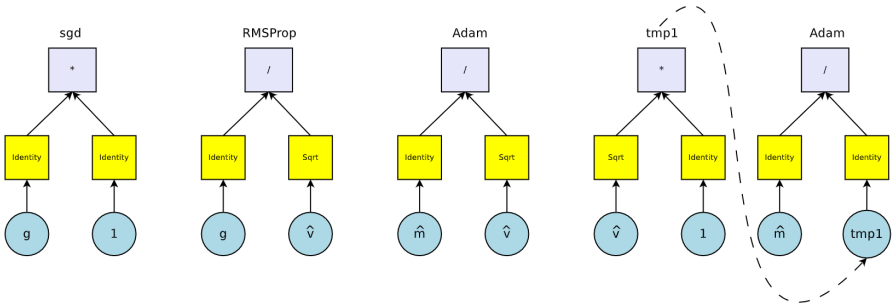


Figure 2.2: Example of using recurrent networks to generate the computation graph of optimizers. From [Bel+17]: here are depicted the computation graphs of (from left to right) SGD, RMSProp, and Adam (entire and decomposed graph). g: gradient — $\hat{m}$: bias-corrected running estimate of the gradient — $\hat{v}$: bias-corrected running estimate of the squared gradient.

posed an approach based on an RNN controller that is trained to generate the computation graph of an optimizer. The operands and the (unary and binary) operators of the optimizer are first modeled in the form of a computation graph (see Figure 2.4). This process is repeated recursively until the full computation

[8] Each component of the learning rate vector corresponding to a weight of the model is divided by an accumulator which aggregates the recent gradients of that weight.

graph of the optimizer is constructed. This process is carried out by the successive outputs of the RNN controller. Figure 2.3 illustrates the process.

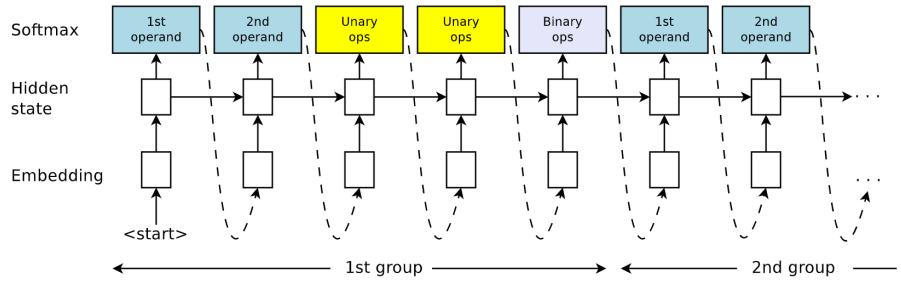


Figure 2.3: Schematic representation of the RNN controller used in [Bel+17] to generate the computation graphs of the optimizers depicted above.

Recurrent models are used again as meta-level model controllers but now to generate the model descriptions of the base-level neural network in a similar way as neural architecture search approaches. For example, Zoph and Le [ZL16] proposed to train a recurrent network as a meta-level model to generate the model description of base-level neural networks. The idea is to predict the values of what we usually refer to in neural architecture search as models' hyperparameters, including, in the case of convolutional neural network models, filter height, filter width, stride height, and the number of filters. Again here, the values of these hyperparameters correspond to the successive outputs of the RNN controller (see Figure 2.4). More broadly, these approaches that generate the models' descrip-

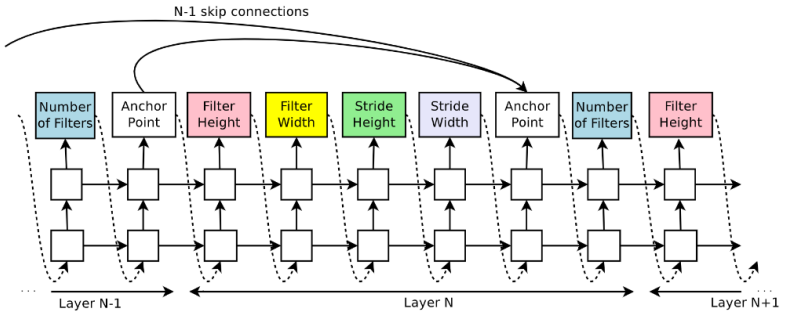


Figure 2.4: An example of using recurrent networks to generate the computation graph of convolutional networks. From [ZL16]: depicted here is an overview of the RNN controller used to sample a simple convolutional network by predicting the hyperparameters of these kinds of models like filter height, filter width, stride height, stride width, and the number of filters for one layer.

tions essentially boil down to the problem of neural architecture search. In the case of the model proposed by Zoph and Le, the problem is typically modeled as a hyperparameter optimization problem where the meta-knowledge corresponds to the hyperparameters (or architecture) being optimized [9]. In a nutshell, neural architecture search works by iteratively sampling architectures from a search space using a given sampling strategy, e.g., Bayesian optimization of hyperparameters or evolutionary algorithms, and training that architecture. The obtained validation performance is used to update the search strategy towards better regions of the search space. According to the taxonomy proposed by Hospedales et al. in [Hos+21], the search space corresponds to the space of meta-knowledge, the search strategy to the meta-optimization strategy, and the performance estimation strategy to the meta-objective.

The learning objective can also be meta-learned in the same way as the parameters of the models or the learning rate. Several works also proposed to meta-learn the objective function ℓ and generally proceed by parameterizing the loss function with a set of learnable parameters [Yu+18; KSS19; Bec+21]. For example, authors in [Bec+21] proposed a framework where both the model's parameters and the meta-loss (a parameterized loss function) are optimized. Similarly, authors in [KSS19] studied this problem in a reinforcement learning setting. They investigated ways of improving a parameterized objective function ℓ_α which is intended, in turn, to improve the policy of the agents. They proposed a differentiable critic which measures the effect of updating the policy as a function of the objective parameters α.

In the same vein as the parameterization which is applied to the optimized loss function, the evaluation strategy for assessing the resulting learning models can also be parameterized and, thus, meta-learned. Indeed, model evaluation based on cross-validation usually relies on a random partitioning process. The random partitioning used in the case of segmented time series introduces a neighborhood bias [HP15]. This bias consists of the high probability that adjacent and overlapping sequences, typically obtained with a segmentation process, which share a lot of characteristics, fall into training and validation folds simultaneously. This leads to an overestimation of the validation results and goes often disregarded in the literature. We investigated in our previous works [OHC17b; OHC17a] the impact of such bias. To alleviate the overestimation problem, various approaches were proposed, such as meta-segmented partitioning [HP15]. The idea is to circumvent this bias by, first, grouping adjacent frames into meta-segments of a given size. These meta-segments are then distributed on each fold. The size of these meta-

[9] Note that in [HKV19], neural architecture search is considered, along with and at the same level as meta-learning and hyperparameter optimization, as an approach of the family of autoML methods.

segments could be learned and contextualized by the type of processing pipeline being used to process the data and time series specifically. Figure 2.5 illustrates an example of the resulting partitioning obtained using random partitioning (often used with regular cross-validation) vs. meta-segmented partitioning [HP15].

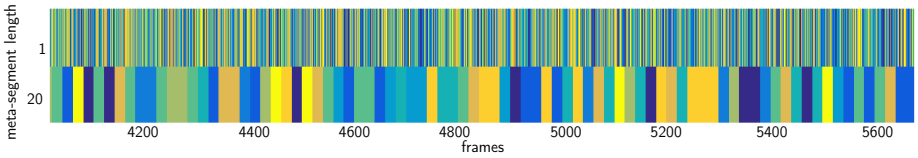


Figure 2.5: Partitioning of a portion of the dataset's frames (or segments) over 10 folds using: (top) regular random partitioning used in traditional cross-validation procedures; (bottom) meta-segmented partitioning algorithm proposed in [HP15]. A segment length of one corresponds to the partitioning produced by the regular cross-validation procedure. The illustrated frames are temporally ordered. Each color corresponds to a different fold.

Other approaches employ a parameterized meta-learner to generate a non-parametric base-level model. Examples of such approaches are [Vin+16; SSZ17; Sun+18] This is often referred to as "learning to compare" [Sun+18] where the idea is to devise an appropriate metric space to perform comparisons between examples and make better predictions. Figure 2.6 illustrates the common process for learning a metric space. Basically, as authors in [HRP21] frame it, the idea of metric-

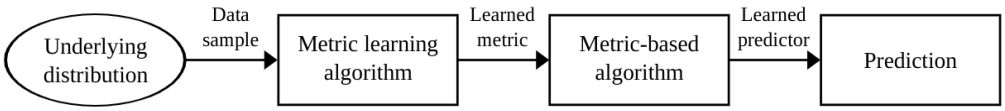


Figure 2.6: Overview of the metric learning pipeline. From [BHS13]: the principle of this pipeline is to devise a genuine metric from a given data distribution that can be used by a learning algorithm to output a predictor (or non-parametric base-level model) which is more adapted to the considered data distribution.

based approaches is to learn meta-knowledge in the form of an appropriate feature (or metric) space where new inputs (that we want to classify, for example) can be compared efficiently with examples (which corresponding true label is already known), in other words, the higher similarity between a new input and a known example w.r.t. the learned feature space the more likely they will get the same label.

What emerges from all the approaches that we reviewed above is that, in contrast to traditional machine learning, meta-learning consists of two nested learn-

ing problems that are often referred to as an inner loop and an outer loop (or a meta-level and a base-level) [Sch87; FAL17]. On the one hand, the inner loop (or base-level) refers to the regular learning setting, such as supervised learning. It solves a specific task (or learning problem), which is determined by a given objective, a data distribution, a hypothesis space, and a learning algorithm. On the other hand, the outer loop (or meta-level) is rather concerned with, as Thrun and Pratt put it, "learning properties of functions", i.e., finding meta-knowledge that makes it easier to solve the base-level problems. Framed differently, the idea is to solve a meta-problem at a higher level in order to put the learner in better conditions to solve basic-level problems. For example, as we reviewed above, meta-level models like the RNN controllers are used to provide appropriate learning rates, computational graphs of optimizers, and architectures of convolutional networks, able to solve base-level problems efficiently. From another perspective, the outer loop consists of learning inductive biases from data, also referred to as "data-driven inductive bias" in [Jer+19] or "inductive bias learning" in [Bax00].

We saw in this section different examples of meta-learning approaches. Gradient-based meta-learning approaches are probably among the most representative approaches, which clearly implement this nested bi-level learning problem. We take a closer look at this type of approach in the following.

2.3 Gradient-based meta-learning

These are probably among the prominent approaches which recently revitalized the meta-learning community. Gradient-based meta-learning methods aim to learn inductive biases in the form of an appropriate initialization so that the learner can adapt rapidly, i.e., within a few gradient steps, to a new task. From an optimization landscape perspective, GBML approaches try to find meta-parameters that lie within a few SGD steps from a wide range of task-specific minima, i.e., these meta-parameters are optimized for fast adaptability to new tasks within a few gradient steps. Formally, we consider in this setting a collection of T tasks τ, indexed by i, drawn from a task distribution $\rho(\tau)$. Each task τ_i has an associated dataset $\mathcal{D}_{\tau_i} = \{(x_j, y_j)\}_{j=1}^{n_{\tau_i}}$ from which we sample two disjoint sets: $\mathcal{D}_{\tau_i}^{train}$, used to fit a model on task τ_i and $\mathcal{D}_{\tau_i}^{test}$, used to evaluate how well this adapted model generalizes on that task.

GBML approaches are often cast as a bi-level optimization problem:

$$\min_{\theta^{(0)}} \mathbb{E}_{\tau}[\ell\big(\theta_{\tau_i}^{(L)}, \mathcal{D}_{\tau_i}^{\text{test}}\big)] \tag{2.5}$$

$$\text{s.t. } \theta_{\tau_i}^{(t+1)} = \theta_{\tau_i}^{(t)} - \eta \nabla_{\theta} \ell(\theta_{\tau_i}^{(t)}, \mathcal{D}_{\tau_i}^{\text{train}}) \qquad \theta_{\tau_i}^{(0)} = \theta^{(0)} \text{ and } \forall \tau_i \sim \rho(\tau), \tag{2.6}$$

where the inner-loop solves a specific task, here by adapting task-specific parameters $\theta_{\tau_i} \in \Theta$ (or θ_i) by minimizing a loss function $\ell\big(\theta_{\tau_i}; \mathcal{D}_{\tau_i}\big)$ using a local optimizer in L steps. Whereas the outer loop optimizes for fast adaptability by aggregating the task-specific gradient steps into a set of meta-parameters $\theta_{\tau_i}^{(0)} \in \Theta$ (or ϕ) used to initialize the task-specific parameters [10]. Figure 2.7 illustrates conceptually how the bi-level optimization process takes place in the parameter space (or optimization landscape).

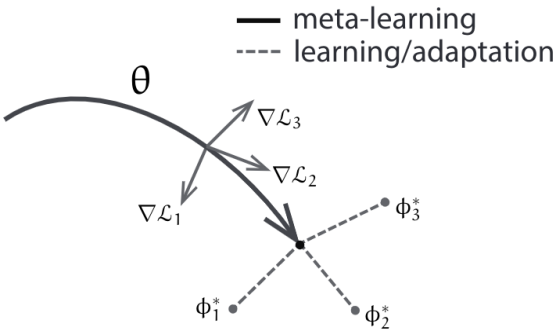


Figure 2.7: Overview of how the model-agnostic meta-learning algorithm proposed by [FAL17] performs the optimization process in the parameter space. The representation ϕ the approach optimizes for, and usually referred to as universal representation, is one that can quickly adapt to new tasks.

[10] Note that θ and ϕ correspond to the same set of parameters, i.e., there is no distinction apart from the fact that they are treated differently in the inner and outer loops (see § 2.4 for alternative configurations).

Algorithm 1: From [Che+19]: Meta-learning pseudocode. TaskAdapt as well as Δ_{τ_i} can take various forms as described in the text.

Input : Size of the task batch B, number of inner loop adaptation steps L, and learning rate η

Initialize ϕ
while *not done* **do**
  $\{\tau_1, \ldots, \tau_B\} \leftarrow$ sample mini-batch of tasks
  for *each task* τ_i *in* $\{\tau_1, \ldots, \tau_B\}$ **do**
    Initialize $\theta_{\tau_i} \leftarrow \phi \ (\equiv \theta_{\tau_i}^{(0)})$
    for *step* $l = 1 \ldots L$ **do**
      $\theta_{\tau_i} \leftarrow \text{TaskAdapt}(\mathcal{D}_{\tau_i}, \phi, \theta_{\tau_i})$
    end
  end
  // Meta update
  $\phi \leftarrow \phi - \eta \cdot \frac{1}{B} \sum_{\tau_i} \Delta_{\tau_i}(\mathcal{D}, \phi, \theta_{\tau_i})$
end

The above procedure, encompassing the original MAML and its derivatives such as implicit MAML (iMAML) and Reptile, is summarized, as suggested in [Che+19], in Algorithm 1, where TaskAdapt executes one step of optimization of the task-specific parameters, and Δ_{τ_i}, referred to as the meta-update, corresponds to the contribution of a task τ_i to the meta-parameters. These contributions are computed, in the case of MAML, by gradient descent on the test loss $\ell_{\tau_i}^{\text{test}}(\theta_{\tau_i}) = \ell(\mathcal{D}_{\tau_i}^{\text{test}}; \theta_{\tau_i})$, resulting in the meta-update $\Delta_{\tau_i}^{\text{MAML}} = \nabla_\phi \ell_{\tau_i}^{\text{test}}(\theta_{\tau_i}(\phi))$. Alternative approaches to MAML were proposed in the literature to alleviate the computation burden that stems from the necessity to backpropagate through the task adaptation process. These approaches rely on the possibility of computing the meta-gradient based solely on the result reached by the inner loop (or adaptation process). Reptile for example optimizes θ_{τ_i} on the entire dataset $\mathcal{D}_{\tau_i}$, and moves ϕ towards the adapted task parameters, yielding $\Delta_{\tau_i}^{\text{Reptile}} = \phi - \theta_{\tau_i}$. Conversely, iMAML introduces an L2 regularizer $\frac{\lambda}{2} \|\theta_{\tau_i} - \phi\|^2$ and optimizes the task parameters on the regularized training loss. Figure 2.8 illustrates the differences between how differentiation is made in MAML, first-order MAML, and iMAML. Besides the burden brought by the cost of the second-order derivative of MAML, the authors in [AES19, §3.1] enumerate other issues such as "*training instability*" and "*Shared Inner Loop (across steps and across parameter) Learning Rate*". Various strategies have been proposed to deal with these issues.

GBML approaches have witnessed tremendous advances in recent years, leading to spectacular results in various applications. Various works have tried to under-

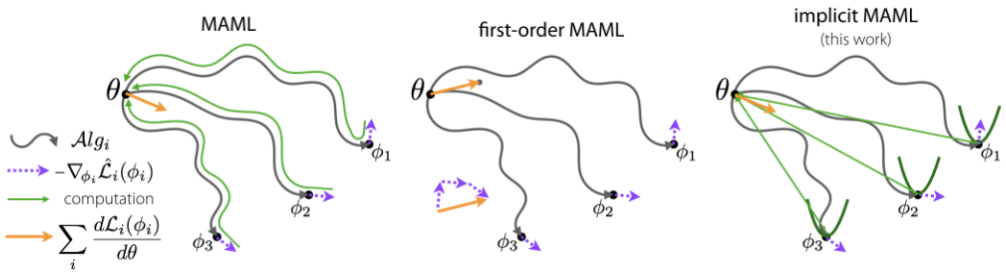


Figure 2.8: Illustration in the parameter space between the way differentiation is performed (from left to right) MAML, first-order MAML, and iMAML. Differentiation through the entire optimization path, as done by MAML, is prohibitive, especially when confronted with long paths: first-order MAML perform an approximation while iMAML derives an analytic expression for the meta-gradient. Figure from [Raj+19].

stand the reasons behind these performances. From a landscape optimization perspective, as we said above, meta-learning approaches try to find meta-parameters, referred to as universal parameters in [FAL17], that lie close to a wide range of task-specific minima. However, the existence of such meta-parameters depends entirely on the curvature of the optimization landscape, i.e., task-specific minima would not necessarily lie close together, and consequently, no meta-parameters would satisfy the within-few-SGD-steps closeness that such approaches rely on. Authors in [Fle+19] advocate for a view where meta-learning corrects the natural ill-conditioned curvature of the optimization landscape over the distribution of learning problems (or tasks) (see the top row in Figure 2.9) [11]. These corrections are intended to prevent gradient descent from struggling in low-interest regions of the optimization landscape (see black curves in Figure 2.9), thus facilitating learning [12]. More formally, Flennerhag et al. suggest materializing this inductive bias in the way the meta-learned update rule is performed, i.e., $\theta_{\tau_i} = \theta_{\tau_i} - \alpha\Omega_\phi(\theta_{\tau_i})\nabla\ell(\theta_{\tau_i})$, where an additional projection, Ω_ϕ, is introduced. This projection operator is parameterized by ϕ, which controls how the projection corrects the optimization landscape curvature.

Performances of meta-learning approaches can also be interpreted from the

[11] This view can be equivalently seen as metric space learning or feature learning. The difference is the space where meta-learning is performed, i.e., parameter optimization space versus feature space.

[12] Just as the order of the learning examples has an impact on the loss surface (or optimization landscape), the order of the tasks that the learner (or the system more generally) encounters can possibly influence the loss surface (as shown in figure 2.9 and explained by so-called "curriculum learning" approaches.)

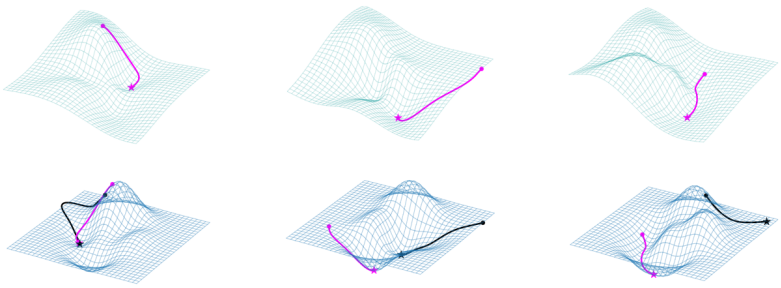


Figure 2.9: Illustration of how meta-leaning acts on the curvature of the optimization landscape following the interpretation of Flennerhag et al. in [Fle+19]. From [Fle21]: the bottom row illustrates the optimization landscape defined by each individual task, while the top row shows how the optimization landscape is got corrected by leveraging the regularities across tasks. Gradient descent is somehow facilitated after the meta-learning process corrects the optimization landscape, also called "meta-geometry".

perspective of feature learning (or the feature space, in contrast to the optimization landscape). The learner, by encountering new tasks, e.g., new positions (data sources), new instances of sensors, new classes, or new contexts, can easily adapt, as a side effect, due to the generalization capabilities acquired during the learning process. Besides acting on the curvature of the optimization landscape, meta-learning approaches are known to optimize for more general features by, again, leveraging the regularities across tasks. Feature learning is a core component of the learning process. As Vapnik frames it in [Vap95]: "Real-life problems are such that there exists a small number of "strong features," simple functions of which (say linear combinations) approximate well the unknown function. Therefore, it is necessary to carefully choose a low-dimensional feature space and then use regular statistical techniques to construct an approximation." The question shifts then to find these sets of strong features. In meta-learning, this is what is referred to as the bias learning problem (as illustrated in Figure 2.10) [Bax00]. This is formalized by Baxter as follows. The set of "strong features" may be viewed as a function $f : X \to V$ mapping the input space X into some (typically lower) dimensional space V. Let $\mathcal{F} = \{f\}$ be a set of such feature maps (each f may be viewed as a set of features $(f_1; \ldots; f_k)$ if $V = \mathbb{R}^k$). In general, the "simple functions of the features" may be represented as a class of functions $\mathcal{G}$ mapping V to Y. If for each $f \in \mathcal{F}$ we define the hypothesis space $\mathcal{G} \circ f := \{g \circ f : g \in \mathcal{G}\}$, then we have the hypothesis space family

$$\mathbb{H} := \{\mathcal{G} \circ f : f \in \mathcal{F}\} \tag{2.7}$$

The problem of "carefully choosing" the right features f is equivalent to the bias learning problem "find the right hypothesis space $\mathcal{H} \in \mathbb{H}$."

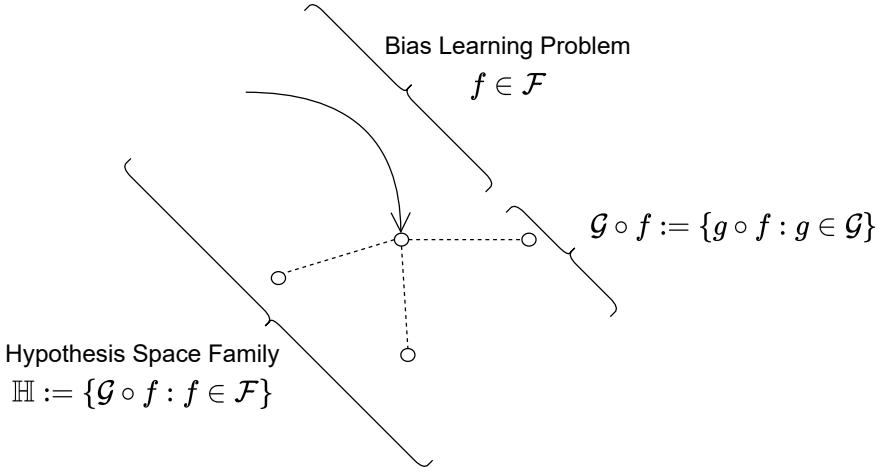


Figure 2.10: Feature learning corresponds to the bias learning phase (or meta-learning), where one seeks to find a feature map (or the right hypothesis space $\mathcal{H} \in \mathbb{H}$).

Often in the context of classical learning, what we try to build is a set of discriminatory characteristics (or patterns) in order to be able to distinguish between different classes. For example, to distinguish between running and walking activities, we will try to focus on the frequency of foot movements: higher frequency in the case of running than walking. It is the value of the frequency that is learned and not the fact that the frequency is a discriminatory characteristic of the learning problem in question. Indeed, this must be distinguished from the goal of meta-learning, which, unlike the construction of discriminatory feature sets in the case of the base machine learning (which is equally important), rather seeks generic (or invariant as we will see in Chapter 6) in order to generalize to other activities: if we take the example of the distinction between the activities running and walking, in conjunction with the learning of discriminatory characteristics (or patterns)—e.g., frequency of movements, the learning process tries to capture generic aspects structuring these activities [13]. The latter, being by definition generic and therefore applicable to structurally similar activities, would make it possible to obtain multiple properties, including better generalization (which is the primary objective of learning) and, by side effect, easier adaptation to incorporating new instances of training configurations—e.g., a new class in the case of the illustrative example.

[13] Vilalta and Drissi, for example, point out to the existence of patterns in each domain and across domains. Noticeably, across domains patterns are invariant in nature.

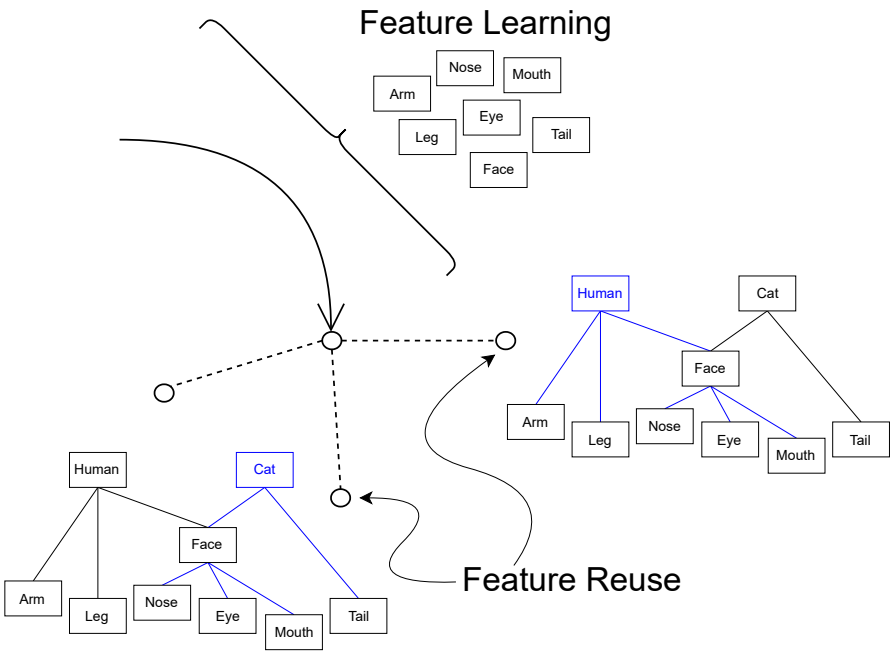


Figure 2.11: Illustration of the concept of "feature reuse" as opposed to "rapid adaptation" discussed, for example, in [Rag+19; Gol+20]. Here we contextualize the process with the way meta-learning (Figure 2.7) performs optimization in the parameter space. The features are learned during the feature learning phase (meta-update) depicted here and then used in each task with different weighting during the feature reuse phase (task adaptation). The meta-learned features are found by Goldblum et al. in [Gol+20] to be qualitatively different from conventional features, which makes them especially appealing for few-shot learning.

On the other hand, this distinction makes links with the aspects of data sketching and modular neural networks (§ 2.4). Indeed, modules such as "Arm", "Leg", and "Eye", illustrated in blue in Figure 2.11, are typically learned elements in the context of meta-learning because these are aspects common to all living beings. On the other hand, what will make it possible to distinguish between the classes "human" and "cat" (always as illustrated in the same figure) is the contribution (or activation) of each of the modules characterized by the weights assigned to the links that connect the different modules (arrows in the figure), and this according to the class to which the input image belongs [14]. Some works, moreover, are inter-

[14]Note that by extension, these remarks illustrated on classes (activities, more precisely) apply similarly (under some extra assumptions), for example, to the notion of tasks, perspectives

ested in this question—e.g., [Rag+19; Gol+20]— whose first results go much more in the direction of what they call in [Rag+19] "feature reuse", i.e., a dictionary of high-level features reused in each task (or domain), rather than features that are optimal for what they call "rapid learning" (or rapid feature adaptation). Authors in [Rag+19] investigate the fundamental strength of MAML and ask specifically, as they framed it: "is the effectiveness of MAML due to the meta-initialization being primed for rapid learning (large, efficient changes in the representations) or due to feature reuse, with the meta-initialization already containing high-quality features?" They find out that feature reuse is mainly behind the effectiveness of MAML approaches. Furthermore, while investigating this question, the authors have also pointed out the distinction between different layers of neural networks with regard to the learned features. Indeed, the earlier layers (the body of the network) do not change drastically during the adaptation phase, in contrast to the head (final layer) of the network. This suggests broadly that the body of the network encodes the features, and the head is responsible for combining them depending on the task at hand. We will discuss these aspects further in the next section (§ 2.4).

By extension, these aspects are linked, among other things, to neural architecture search approaches and the integration of domain knowledge into the learning process. Indeed, a growing literature is taking place on training-free neural architecture search, which searches for structures that do not need full training on data to match the data distribution (or solve the problem at hand) [CGW20; Lin+21; Mel+21; Xu+21b; Wan+22]. The idea is that architectures already encode inductive biases per se. Weight adaptation becomes, therefore, trivial in comparison with the architecture search process, which looks for suitable architectures encoding precise inductive biases, e.g., weight-agnostic neural networks [GH19]. Figure 2.12 illustrates a perspective on the neural architecture search interpreted as a bias learning problem followed by a weight adaptation phase allowing the neural architecture to tackle specific tasks.

Overall, we reviewed in this section the popular GBML approaches with a special focus on the various intuitive interpretations, i.e., optimization landscape and feature learning perspectives. The debate around feature reuse and rapid learning in [Rag+19; Gol+20] led to an interesting question regarding how aspects of the problems (task-specific and domain-agnostic) are captured and by which portions of the model's parameters.

(views), sensor instances (with particular characteristics), position of the data generators in the space, or (geographical) contexts.

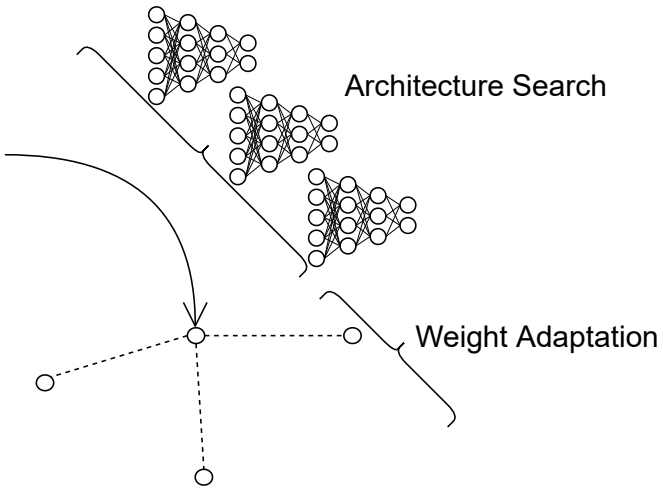


Figure 2.12: Neural architecture search can be seen as a bias-learning process where one looks for the most appropriate way of tying the neurons together (as opposed to learning their values). This defines what an architecture is and encodes a particular inductive bias per se. Weight adaptation is sometimes unnecessary because the obtained architectures already encode some biases targeted toward the learning problems. This is particularly the case in the works around weight-agnostic neural networks, e.g., [GH19].

2.4 Parameters and meta-parameters

Thrun and Pratt evoke in [TP98] the notion of functional decomposition as a design for the construction of learning-to-learn systems. Basically, this design consists in building functions of the form $h = f_i \circ g$, where f_i is task-specific whereas g is shared across all h_is. In this section, we take a look at approaches that make an explicit distinction between parameters and meta-parameters, i.e., partitioning the parameter space according to the aspects of the problem (or data) one is aiming to capture [15].

In the case of MAML, the meta-parameters ϕ are used to initialize task-specific parameters θ, meaning that they both correspond to the same set of parameters. This restricts the capabilities of these kinds of approaches when confronted with relatively dissimilar tasks, as we discussed previously from an optimization landscape perspective. Following the spirit of the functional decomposition approach

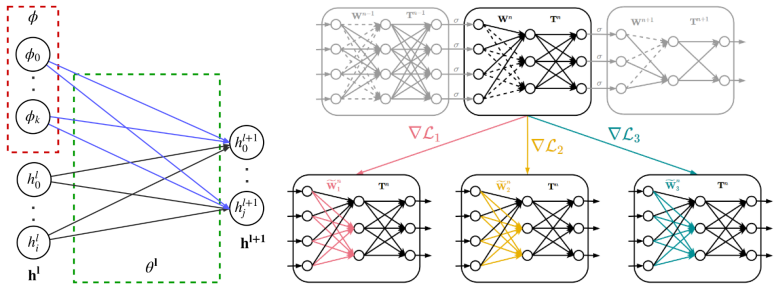
[15]Note here that meta-parameters are not to be confused with hyperparameters. Indeed, in this context, meta-parameters are special parameters that capture specific aspects of the learning problem (or data distribution). A hyperparameter here could be the way the space partitioning is performed between parameters and meta-parameters as well as how each of them is learned.

in [TP98], a natural alternative to how MAML treats these parameters is to split them into distinct groups, for example, two-group splits with a first group that varies across tasks and a second one that is shared (or invariant) across tasks. Furthermore, these splits can either be explicitly decided a priori or emerge heuristically during the learning process. With this distinction between parameters and meta-parameters, the meta-learning objective becomes:

$$\min_{\theta^{(0)},\Phi} \mathbb{E}_{\tau}[\ell(\theta^{(L)}_{\tau_i}, \Phi, \mathcal{D}^{\text{test}}_{\tau_i})] \tag{2.8}$$

$$\text{s.t. } \theta^{(t+1)}_{\tau_i} = \theta^{(t)}_{\tau_i} - \eta \nabla_{\theta} \ell(\theta^{(t)}_{\tau_i}, \Phi, \mathcal{D}^{\text{train}}_{\tau_i}) \qquad \theta^{(0)}_{\tau_i} = \theta^{(0)} \text{ and } \forall \tau_i \sim \rho(\tau). \tag{2.9}$$

This formulation makes explicit the notion of meta-parameters Φ (supposed to be shared across tasks), which are explicitly optimized for, as opposed to the previous formulation. Following this formulation, MAML does not have any additional meta-parameter, i.e., ($\Phi \equiv \emptyset$), as the met-parameters used to initialize the task-specific parameters coincide.



(a) (b)

Figure 2.13: Two examples of the model partitioning into parameters and meta-parameters. (a) From [Zin+19] an illustration of a network layer with network parameters θ (green) augmented with context parameters ϕ (red). Update procedure during and outside adaptation step makes these two portions of parameters learn task-specific and task-agnostic aspects of the learning problem. (b) From [LC18] another network layer featuring an initial set of weights (black arrows) and weights to be updated (dotted arrows) by task-specific learners. The weights to be updated are determined by the meta-learner.

Various approaches have been proposed in this direction, e.g., [Zin+19; Rag+19; Che+19; LC18; KKB18; KBT19]. For example, in ANIL [Rag+19], the explicit distinction between parameters and meta-parameters arises in neural networks through the layers, and their depth in the network, e.g., the last layer corresponds to the parameters (or task-specific parameters), θ, while the shared embedding

(front-end layers) corresponds to the meta-parameters, Φ [16]. In [Zin+19], authors proposed CAVIA (for fast context adaptation via meta-learning), which augments each layer of a given model with additional parameters, referred to as context parameters, and adapted in the inner loop for each task (see Figure 2.13a). Here the partitioning (which actually corresponds to the augmentation) is hand-crafted, and the training rule is adapted accordingly, i.e., the network parameters associated with the context parameters (depicted in blue in Figure 2.13a) are disabled, i.e., set to 0, before adaptation. Therefore, these parameters do not affect the output of the layer. After that, they are enabled, allowing them to modulate the output of the layer depending on the task at hand. This is what makes the two portions of parameters capture different aspects of the learning problem. Other approaches, such as MT-Nets [LC18], have also proposed to learn this partitioning automatically: the meta-learner specifies a mask indicating which parameters to update in the inner loop and the task-specific learner is responsible for updating the parameters in question (see Figure 2.13b). This approach has the advantage of coming up with adaptable modules via the binary mask variables, while CAVIA has the advantage of being simpler and more interpretable, as the parameters and meta-parameters are disjoint sets.

Beyond determining this important question of model parameter partitioning, there is the equally important question of the way each parameter (or partition) is updated according to this partitioning. This question can boil down, for simplicity, to the choice of the appropriate learning rate for each parameter (or partition). Indeed, shared parameters (or meta-parameters) tend to evolve slowly (or stay quasi-invariant) across tasks, while task-specific parameters are volatile. This observation has to be reflected in the learning rate, as it is responsible for controlling the parameter updates. In [KBT19], authors investigated the notion of parameters and meta-parameters from this perspective and devised algorithms that provably adapt to task similarity and to dynamic environments by using suitable update rules for the meta-initialization $\phi \in \Theta$ and the learning rate $\eta_{\tau_i} \in \mathbb{R}$. As highlighted by [AES19, §3.1], where authors provide best practices for training MAML models, using a shared learning rate for all parameters without distinction and all update steps has a major impact on the generalization and convergence rates: some of the model parameters are consistent across tasks, like the feature extractors, while others are task-specific, like the classification layers (see Figure 2.14). Therefore gradient updates must stay consistent regarding this structure. In this sense, authors in [KBT19, §4] also consider adaptation to a more sophisticated task-similarity structure by learning a per-coordinate learning-rate $\eta \in \mathbb{R}^d$ in or-

[16] The distinction between the shared embedding and the last layer has been observed empirically in various works and corresponds, for example in multi-task learning, to a fundamental construction, e.g., in MTL, the last layer (or head) is usually added on top of the shared embedding and fine-tuned to match a new task.

der to get iteration $\theta_{\tau_i}^{(t+1)} = \theta_{\tau_i}^{(t)} - \eta_{\tau_i} \odot \nabla_{\tau_i}^{(t)}$, which is more practical and can leverage multi-task information to adapt the within-task learning rate. These as-

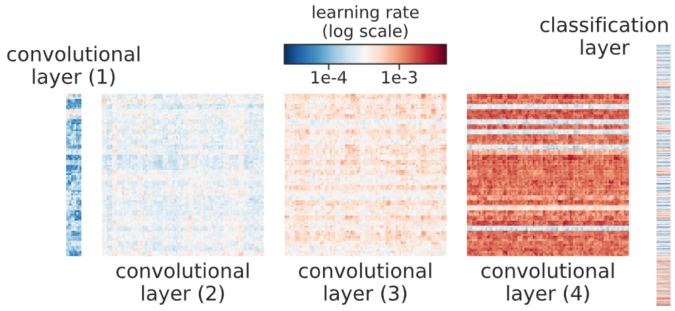
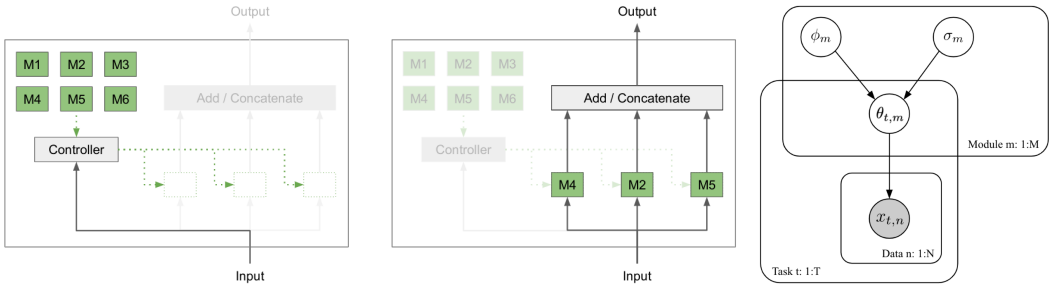


Figure 2.14: Variation of the learning rate across layers of a convolutional neural network trained on a famous image recognition benchmark (from [KBT19]). As the lower-level layers tend to be frozen (share across tasks), the corresponding per-coordinate learning rates are smaller compared to those of the layers which are closer to the output (task-specific).

pects are investigated in the continual learning setting, e.g., [GYP20], as well as in multi-task learning settings, e.g., [Yu+20], where the idea is to minimize *gradient interference*, i.e., the gradients for different tasks point away from one another. This enables a more adaptable and efficient way to mitigate catastrophic forgetting in the continual learning setting and better generalization overall.

Modularity and data sketching are two notions that participate in pushing further the parameter partitioning strategy pursued in the meta-learning literature, e.g., [KKB18; Che+19]. Indeed, the meta-parameters (or universal parameters) shared by all the tasks (domain, contexts, etc.) correspond to reusable (and potentially interpretable) modules. These approaches go a little further than the simple distinction between parameters and meta-parameters of models. This family of approaches is designated in [TP98], analogously to the notion of functional decomposition of learning-to-learn systems, as "*piecewise function decomposition approaches*" which models functions h_i as a collection of functions $f_1, f_2, ..., f_M$ corresponding to the notion of module of data sketch. Figure 2.15 illustrates the principle via the architecture of the modular layer proposed in [KKB18] and the Bayesian shrinkage graphical model proposed in [Che+19]. Specifically, the model parameters in [Che+19] are partitioned into M disjoint modules $\theta_{\tau_i} = (\theta_{\tau_i,1}, \ldots, \theta_{\tau_i,m}, ..., \theta_{\tau_i,M})$, where $\theta_{\tau_i,m}$ correspond to the parameters in module m for task τ_i and can be materialized by, e.g., feature maps, neural network layers, or any other building block of the learning models. Shrinkage parameters attached to each individual module are learned during the process and



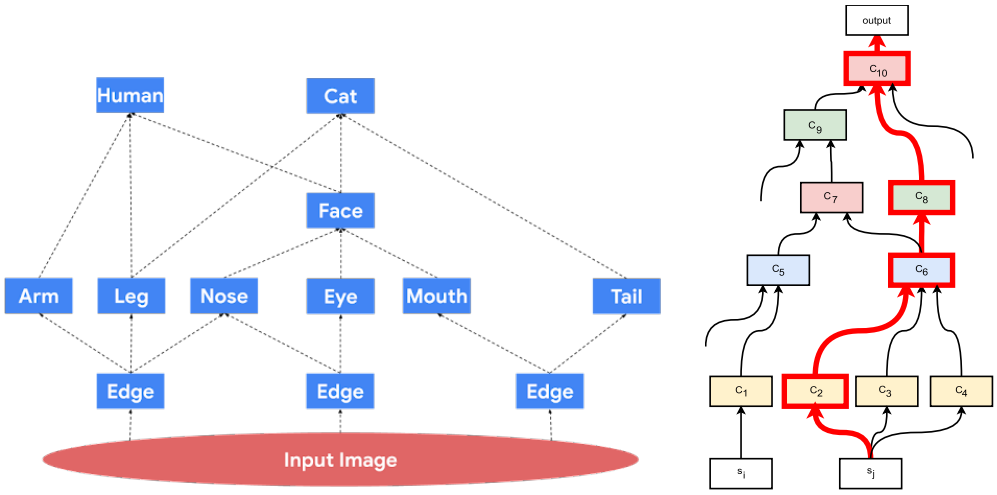
(a) (b)

Figure 2.15: Two examples of modular networks. (a) From [KKB18] an illustration of a modular network where a controller is responsible for selecting a set of modules from the available ones. These are then used to process the input at hand. (b) From [Che+19]: the Bayesian shrinkage graphical model showing the task-specific parameters θ_{τ_i} and meta-parameters ϕ. The shrinkage parameters σ control which subsets of parameters (or modules) to fix and those to adapt and those to adapt for each individual task.

used to determine which modules are task-independent and thus can be reused at test time and which are not, to be adapted for each task. Modularity goes further than simply partitioning the parameter space into basic parameters and meta-parameters.

In this sense, modularity is a key enabler for the development of transparency in machine learning models. Indeed, the transparency of a model is linked to the ability to understand its internal operating mechanisms. In particular, along with "*simulatability*" and "*algorithmic transparency*", the "*decomposability*", i.e., "each part of the model—input, parameter, and calculation—admits an intuitive explanation" as Lipton puts it in [Lip18], is one of three levels around which this notion of transparency is structured. Modularizing a given model and correlating them with human-understandable concepts like in data sketching [GPW19] (see Figure 2.16a) is one way to go about interpretability. This is typically how, in principle, representation-based methods such as network dissection representation [Bau+17] and concept activation vectors characterize portions [Kim+18] of the model's internals [Gil+18].

For example, we investigated in [HO20] the concept of modularity in the model's internals and how the modules correlate with domain knowledge. We were interested in human activity recognition from on-body sensor deployments, and the objective was to have control over the influence of data sources (s_j in Figure 2.16b) on the learning process. The meta-learning phase enabled us to exhibit



(a) (b)

Figure 2.16: (a) From [GPW19]: illustration of a network sketch used to process an input image. A recursive sketching mechanism was proposed in their work. (b) From [HO20]: Highlighted in red is one of the paths that have as source node the data source denoted by s_j, and that is processed by the architectural components C_2, C_6, C_8, and C_{10} before joining the architecture's output node.

modules (C_i in Figure 2.16b) related to domain knowledge, materialized in this case, by the most influential data sources in the on-body sensor deployments w.r.t. the final learner's outputs. Similarly, in Chapter 6, we discuss related aspects from the perspective of group-invariant representations, where the group action of each individual data source is constrained to act on specific parts of the latent space. This allows, in particular, to control the learning process but ultimately to have interpretable representations and assess the impact of each individual data source w.r.t. the domain, e.g., impact of the location in the deployment, sensor characteristics, etc.

We reviewed in this section the distinction between the parameters and meta-parameters that allow meta-learning approaches noticeably to capture task-specific and task-agnostic aspects in the data. Another distinction of similar importance in these approaches is that of data and meta-data that the meta-learning models can learn from. Indeed, meta-learning is not limited to learning from data as in the standard learning setting. Other forms of data are used, and this is what we will see in the next section.

2.5 Data and meta-data

There is a distinction between data and metadata used for learning both at the meta and base levels. In this part, we will see that meta-learning does not merely employ data in the traditional meaning of the term: there is no need to recall what classical data are; we will rather focus on meta-data and give some examples. Data in the classic sense, however, has an impact on the meta-level, as we will see, and therefore the elements used for training at the meta-level are not limited to meta-data. One must also distinguish between what is used for meta-learning, i.e., meta-level inputs, and what is meta-learned, e.g., initialization parameters of the base learner.

Together with the task distribution and the data flow between the meta and base levels, the choice of the meta-objective constitutes the third axis (or the "*why?*") of the meta-learning's design space devised in [Hos+21, §3.2]. Some meta-learning approaches try to conceive optimizers at the meta-level that lead to faster convergence rates for the base learner. For example, GBML approaches often optimize for few-shot learning where data efficiency is a strong requirement, i.e., the model must classify data into one of N possible classes, however, there are only k samples of each class in $\mathcal{D}_\tau^{\text{train}}$. In this situation, data and meta-data coincide, and the objective is fulfilled via the evaluation protocol or learning episode design.

The choice of the data used for the adaptation process is of utmost importance, and in this regard, various works highlighted the problem of sensitivity of the meta-learning approaches to the support data [GFG20; OBT21; Xu+21a; AYS21]. For example, authors in [AYS21] studied this problem in image classification and found that the effectiveness of currently available algorithms is extremely sensitive to the support set used for adaptation. They show, in particular, that there are images that, when utilized for adaptation, may produce an accuracy as low as 4% or as high as 95% on common benchmarks for the classification of few-shot images.

Paying particular attention to the choice of the support set in the case of approaches optimizing for few-shot adaptation has close ties with sample weighting and selection strategies, and beyond, with curriculum learning, where data and meta-data coincide again by taking the form of weighting, selection, and ordering. Indeed, these links go even further to cover the privileged information proposed by Vapnik and Izmailov and distilled knowledge of Hinton, Vinyals, and Dean.

On the one hand, in the privileged information framework, Vapnik et al. in [VI15] make an analogy with the fact that humans learn much faster than machines and illustrate this with the Japanese proverb *"better than a thousand days of diligent study is one day with a great teacher"*. The proposed *learning with privileged information* framework consists in considering training data formed by a collection of triplets $\{(x_i, x_i^*, y_i)\}_{i=1}^n \sim P(x, x^*, y)$, where each (x_i, y_i) is a feature-label pair, and the privileged information x_i^* is an additional supervision term

about the example (x_i, y_i) provided by an intelligent teacher (in our case, the surrogate model) in order to support and guide the learning process. Guiding the learning process can either be linked to the learning examples supplied to the learner or the learning configurations (e.g., the topology of the sensor deployments, characteristics of the sensing devices, etc.) that the learner encounters during deployment. The privileged information can be, e.g., relevant features or sample-dependent relevant features [Lop+15]. The selection of suitable hypothesis spaces can leverage the uncertainty accompanying some configurations of the data acquisition step.

On the other hand, the distillation framework introduced in [HVD15] tries to incorporate knowledge, in the form of class-probability predictions, from high-capacity models into low-capacity models. Rather than training low-capacity, deployment-ready models using the raw (hard) labels, class-probability predictions (soft labels) generated by the high-capacity models are used instead. In contrast to a boosting training strategy where the hard-to-classify examples are weighted so that the learner can focus on them, in this framework, the easy-to-classify examples, in the sense of smooth class membership, are supplied during model training instead. This smoothness in class membership (or class probability predictions) is controlled by using an additional parameter (temperature $\in]0, 1[$), which decides how to soften the class membership.

Another taxonomy categorizes meta-learning approaches according to the type of meta-data leveraged for learning-to-learn, from the most general to the most task-specific [Van19], including meta-data originating from model evaluations, tasks properties, and the internals of the models themselves. For example, in the case of model evaluations, meta-data consist of observations about the way the learning process behaves and how it evolves, e.g., when changing the model's architecture or adding more training data, etc. As a parallel, the meta-data used in the case of neural architecture search to update the hyperparameters of the next neural architectures consists of the evaluations of the previously sampled neural architectures. As running a training process until completion is prohibitive, proxy task performances often more affordable to evaluate are used instead, e.g., fewer data and shorter learning epochs, and down-scaled models. The idea is that starting from these evaluations, the models have to predict the best learning configuration, including the model's architecture and training process, e.g., authors in [Bak+17] model learning curves to predict the final performance of a given model.

Prior machine learning models per se, i.e., the structure of the learned models and their learned parameters, can also serve as meta-data from which the meta-learners can learn. Concretely, the meta-learner is supplied with tasks and their corresponding learned models, which it uses to train a base learner on new un-

seen tasks. For example, the meta-initialization strategy featured by the GBML approaches and neural architecture search can be considered as a form of learning from prior machine learning models. Furthermore, properties or characterization of the tasks is another source of meta-data and consists of, for example, simple scalars (e.g., number of instances, number of classes, number of outliers, etc.) or statistics (e.g., data sparsity, skewness, feature correlation, etc.) (see [Bra+22] and [Van19, Table 2.1] for a more featured list). The idea is to use these characterizations to measure the distance between tasks and decide whether or not information can be transferred from one task to another. This type of meta-data is more related to the notion of task-relatedness discussed in the next section.

Indeed, a common assumption in the meta-learning approaches is that tasks necessarily have to be related to each other with some kind of structure. In GBML approaches, this notion of task relatedness is more often assumed or used to derive tighter learning and convergence bounds rather than used explicitly in the form of meta-data, as we have illustrated in this section. We will see that some works try to model and leverage this notion in order to meta-learn and improve the performance of meta-learning.

2.6 Families of (structurally) related tasks

As Edwards and Storkey frame it: "An efficient learner is one who reuses what they already know to tackle a new problem. For a machine learner, this means understanding the similarities amongst datasets." In this section, we will take a look at task-relatedness, a fundamental pillar behind the learning-to-learn machinery [17]. In the following, we will see (i) the different characterizations of task-relatedness found in the literature; we will see (ii) how taks-relatedness is computed, followed by (iii) a little theoretical digression; finally, we will see (iv) how it is explicitly leveraged to improve meta-learning.

A task is defined by Finn in [Fin18] as an entity being learned or adapted to, which could take the concrete form of an objective, domain, environment, or any combination of these. Authors in [ES16], for their part, suggested a task as corresponding simply to the notion of a dataset that could be materialized by the pictures or speech recordings of a particular individual or a given document represented in the form of bag-of-words. In robot learning, authors in [Nag+20]

[17] Alternative views exist on the principle of task-relatedness, e.g., [PL15; SLS21], where the evaluation of meta-learning approaches goes beyond this principle. Noticeably, authors in [SLS21] point out the differences between tasks sampled in in-distribution and out-of-distribution configurations, e.g., the few-shot learning where the test task can substantially differ from the training tasks versus federated meta-learning approaches where clients are considered as tasks (see Section 3.6 for additional details about this last point).

considered scenarios where complex robots get damaged and have to adapt their dynamics rapidly so as to pursue their work. The notion of task here is materialized by the different scenarios, i.e., robot damage. Likewise, authors in [HO21b] studied on-body sensor deployment for human activity recognition, where the consequences of sensor failures and deployment re-configurations have been considered as generating new tasks, which the activity recognition models have to adapt to, again rapidly. More broadly, the notion of task can have a wide spectrum of interpretations and vary according to the type of object being considered in a picture as well as the lighting conditions under which that picture was taken, and even the objective being pursued by a given task.

The need for tasks to be sampled according to a certain distribution $\rho(\tau)$ is a key assumption in the learning-to-learn environment. Indeed, there must be some link between observed activities and future unobserved tasks for meta-learning to occur in a lifelong fashion. This kind of distribution is known as a task family or task environment and was formalized for the first time by Baxter in [Bax00]. Following that, a huge body of work focused on describing and understanding the nature of these task distributions and any structures that connect these tasks together, as well as the conditions under which meta-learning can occur. Various approaches were proposed in the literature to measure task similarity, e.g., [Zam+18; Ach+19; NDC21; Jia+18; TNH19; Kum+21]. These approaches often proceed by computing similarity scores between tasks either by modeling their data-generating process or leveraging semantic information in the label space. For example, Taskonomy [Zam+18] proposes a computational approach for modeling the structure of the space of computer-vision tasks, such as texture recognition, semantic segmentation, reshading, colorization, etc. Task2Vec [Ach+19] projects (or embed) the parameters of a task-specific model into a (lower-dimensional) latent space, abstracted from information regarding the number of classes and class label semantics contained in that given task. Further details are provided in the literature overview in Chapter 5, where we investigate ways of organizing (or structuring) the learning process by leveraging the semantics of the label space and measuring how atomic concepts and group of concepts related to each other. In this context, the atomic concepts and group of concepts are considered to be tasks. The goal being to leverage the structuring of concepts in order to maximize sharing and transfer among these learning tasks.

Indeed, the idea is that after exhibiting the notion of task-relatedness, one can leverage this prior knowledge in order to improve the learning process. For example, one can perform task-clustering in order to devise, in the case of GBML approaches, task-cluster-specific initialization rather than a unique initialization for all tasks, which could be inefficient when tasks are slightly distant from each other [Yao+19]. In the presence of strong dissimilarities among the tasks, find-

ing, for example, a universal optimization in the case of GBML approaches, becomes less likely, and even cluster-specific initializations provided by the above approaches would not be appropriate. A better understanding of the underlying structures is of utmost importance, and such structures could be as simple as the laws of physics, e.g., gravity or Newton's first law of motion, or simple priors used in robot learning, e.g., temporal coherence prior, meaning that task-relevant properties of the world change gradually over time or proportionality prior [Kau20] (see Chapter 4 for more details). In summary, we can say that there are different levels of similarities ranging from the most basic—e.g., related to the basic laws of physics—to the most advanced—e.g., the semantics of the label space—and which are therefore more specific and could potentially lead to better generalization and convergence rates.

Various works have investigated the notion of task relatedness, trying to characterize it and analyze it theoretically by providing, for example, information-theoretic lower bounds on minimax rates of convergence. In particular, this notion is studied from various perspectives, including multi-task learning [Bax00; BB08], domain adaptation (or transfer learning) [Ben+10], and meta-learning [Luc+20; JS21]. One of the first and most important studies to provide theoretical guarantees on generalization and convergence rates for learning-to-learn (and also multi-task learning) was carried out by Baxter in [Bax00], where the first sample complexity bounds under the framework of VC theory [Vap95] have been proposed. As mentioned previously, the predictors h considered here factorize as $h = g \circ f$ and f^* is a feature representation shared across tasks, learned using n samples from each of the T tasks, and adapted to a target new task using n_τ samples from that task. The primary issue being investigated is determining how many samples, denoted by n, are required to learn f^*, which may then be customized for a new task that has not been seen before. A large body of theoretical studies following the work of Baxter provided tighter bounds while still assuming a common generative model over tasks, referred to as task environment, from which tasks are sampled IID. Authors in [MPR16] for example provided generalization bounds scaling to the order of $\mathcal{O}(1/\sqrt{T}) + \mathcal{O}(1/\sqrt{n_\tau})$. One major concern related to this bound, highlighted and studied for example in [TJJ20; Du+20], is that the first term, i.e., $\mathcal{O}(1/\sqrt{T})$, decays only in the number of tasks T but not in n. This does not seem to match empirical evidence in the literature about the practical efficacy of these approaches, particularly transfer learning, where very often $n \gg T$ [TJJ20]. For Du et al., the IIDness of the tasks generative model which is often assumed is not sufficient alone to explain the practical efficacy of these approaches, and the connections between tasks have to be investigated.

Although being a challenging problem, particularly when the parameter space of models like neural networks tends to be quite large and complex, modeling

and leveraging relationships (or structure) among tasks has the potential to improve learning-to-learn by, for example, "fast-track" learning, as Nguyen, Do, and Carneiro put it in [NDC21], of similar tasks by devising more appropriate cluster-specific rather than single initializations or detecting outlier tasks which need much more adaptation efforts [Jer+19]. When the task distribution is heterogeneous, relying on single parameter initialization in the case of GBML approaches was demonstrated empirically to have limits [Vuo+19], the reason is that, as discussed in the optimization landscape interpretation, few gradient steps are not likely going to lead the parameter initialization towards task-specific parameters that satisfy a wide range of tasks.

Multi-task learning (MTL) is perhaps one of the research areas where the notion of structure is leveraged to a greater extent due to the way this problem is formulated. In the *hard parameter-sharing* formulation of MTL problems, i.e., $\min_{W,\Lambda} \sum_{i=1}^{T} \sum_{j=1}^{n_{\tau_i}} \ell_\tau(\theta_\tau^\top x_j, y_j) + R(W, \Lambda)$ [18], one can notice the additional regularization term R, which is actually responsible for imposing this notion of structure. The matrix $\Lambda \in \mathbb{R}^{T \times T}$ (containing task pairwise scalars) is intended to model the structure of the tasks either a priori or while being estimated during the learning process via the regularization term [Smi+17]. The idea is to control how information is shared amongst tasks, i.e., how to bring the weight vectors for each task, θ_{τ_i}, closer to one another when the tasks are similar to one another and farther apart when the tasks are dissimilar to one another. For example, many works assume that the matrix reflects a clustered structure, e.g., [ZCY11; ZY12; JVB08; EP04], and try to impose that during the learning process. Similarly, probabilistic priors can be used to model the dependence among the columns of W, which has the advantage of capturing both positive and negative relationships among the tasks, which are rather difficult to achieve only with clustering. In each case, the chosen structure generates a regularization term in which the general principle is to impose parameter-closeness of the task-specific weight vectors to optimal task parameters.

Still, regarding approaches that rely on additional regularization terms to impose tasks structures, this time closely related to the online learning setting where the task-environment changes dynamically, authors in [KBT19] leverage the geometric structure of the tasks using online mirror descent with a regularizer based on the Bergman divergence, i.e., $\frac{1}{\eta_\tau}\mathcal{B}_R(\cdot||\phi)$ for initialization $\phi \in \Theta$ and learning rate $\eta_\tau > 0$, to impose the notion of parameter closeness as a materialization for task-relatedness. When the number of tasks $T \to \infty$, the average regret scales with V, where $V^2 = \frac{1}{T}\sum_{i=1}^{T} \mathcal{B}_R(\theta^*_{\tau_i}||\phi)$ and when $\phi = \frac{1}{T}\theta^*_{1:T}$, this means that average

[18] In the hard parameter-sharing formulation of the multi-task learning, each task has its own weight vector, i.e., $W := [\theta_{\tau_1}, ..., \theta_{\tau_T}] \in \mathbb{R}^{d \times T}$ is a matrix whose i-th column is the weight vector for task τ_i [Smi+17]. The joint learning process is responsible for constraining the whole network in a way that it leverages commonalities amongst tasks.

performance improves with task-similarity [KBT19]. Ideally, in GBML approaches that optimize for few-shot adaptation objectives, the learnt initialization ϕ should be close to the optimal model parameter $\theta^*_{\tau_i}$ of any task $\tau_i \sim \rho(\tau)$, i.e., small distance $\mathbb{E}_{\tau_i \sim \rho(\tau_i)}[\left\|\phi - \theta^*_{\tau_i}\right\|_2^2]$.

In the same spirit as the MTL approaches, which impose cluster structure to the task-specific weight vectors, learning multiple initializations, $\{\theta^*_c\}_{c=1}^C$, for C groups (or clusters) of tasks rather than for all tasks at once is one way to further reduce this distance. With a better targeted initialization, $\theta^*_c \in \{\theta^*_c\}_{c=1}^C$, for a specific task τ_i belonging to cluster $c \in [C]$, the optimal parameters for each task (or task-specific minima) would satisfy the within-few-SGD-steps closeness discussed in Section 2.3. Various works have been proposed in this sense, e.g., [Jer+19; Zho+21b], where the idea is to learn task-specific cluster assignments and model parameters in a joint fashion. Similarly, approaches leverage task embeddings, i.e., representation of the tasks in a latent space (e.g., Task2Vec [Ach+19]), in order to modulate (or bias) the parameters of the base learners towards good initialization for solving a given target task. For example, authors in [Vuo+19] propose a framework where a *modulation network* produces modulation vectors, σ_i which are applied to each building block (or layer), θ, of a *task network* so as to get suitable initialization parameters. The process is formalized as a modulation operation $\phi = \theta_i \odot \sigma_i$, where ϕ_i is the modulated prior parameters for the task network, and $\odot$ represents a general modulation operator taking various forms, including attention-based modulation and feature-wise linear modulation. Using a well-crafted latent space where similar tasks are projected into the same regions is a way of implicitly incorporating task structure into the learning process.

We saw in this chapter a summary of the relevant meta-learning background with a focus on the task-relatedness, a fundamental aspect at the basis of meta-learning machinery. This is one of the key components that we use to make a junction with the federated learning models presented in the next chapter.

Chapter 3

Federated learning models

In this chapter, we describe federated learning (FL) models and review strategies proposed in the literature to take into account the impact of heterogeneity across clients. A duality between task-relatedness and client heterogeneity is described, which ultimately lays down the principles upon which we build different approaches presented in the following chapters 4—6.

The standard machine learning setting is usually conducted in a unique centralized site where a representative sample of the overall data distribution is available. Conversely, the distributed optimization (or federated learning) setting was described in [KMR15] motivated by an increasingly spreading learning scenario consisting of a large number of mobile devices (also called clients) generating and holding training data locally instead of being aggregated into a unique centralized site. The goal is still to learn a unified theory while conciliating the diversity of clients in terms of the quantities of training data each individual client holds and the representativeness of the training samples of each client regarding the overall data distribution of the whole population. The general description of the federated learning setting was popularized by McMahan and Ramage in [MR17], while its theory was laid down in [Kon+16; McM+17]. This setting fits equally well with the IoT ecosystem and was often relied upon in the literature. Existing challenges, including communication efficiency, heterogeneity of data distributions across clients, and local objectives inconsistency, are obviously interesting but are pushed further by the learning scenarios introduced by IoT applications. This is what we focus on in this chapter.

The rest of this chapter is organized as follows. We first present one common federated learning setting in Section 3.1 before turning into the fundamental challenges surrounding this learning setting, which are the induced heterogeneity

and objective inconsistency among clients (§ 3.2): the various domain specificities presented in the introductory chapter induce heterogeneous components into the FL setting which brings diversity, beneficial for the learned theories, but impairing the learning performances when they are dismissed or not explicitly handled. Section 3.3 provides a quick picture of the different datasets used to study the heterogeneity aspects arising in FL settings, both naturally and artificially. This is followed in Section 3.5 by the various approaches specifically devised to mitigate these heterogeneity aspects. Along with the presentation of these approaches, we try to draw parallels with the learning-to-learn approaches that we reviewed in the previous chapter. These parallels are materialized by the duality between the clients participating in the FL setting and the tasks. This is discussed in Section 3.6, allowing us to make a case for leveraging the notion of structure at various levels amongst tasks and clients for both learning-to-learn and federated learning approaches.

3.1 Federated learning setting

In a common formulation of the decentralized machine learning setting, a set of M clients, each corresponding, for example, to a sensor in an IoT deployment, aim to collectively solve the following optimization problem:

$$\min_{\theta \in \Theta} \left\{ f(\theta) := \sum_{i=1}^{M} \alpha_i \cdot f_i(\theta_i) \right\}, \tag{3.1}$$

where $f_i(\theta_i) = \frac{1}{n_i} \sum_{\zeta \sim P_i(x)} \ell_i(\zeta; \theta_i)$ is the local objective function at the i-th client, with ℓ_i the loss function and ζ a random data sample of size n_i drawn from locally stored data according to the distribution of client i. At each communication round r, each client runs independently T_i iterations of the local solver, e.g., stochastic gradient descent, starting from the current global model (set of weights) $\theta_i^{(r,0)}$ until the step $\theta_i^{(r,T_i)}$ to optimize its own local objective. Then the updates of a subset of clients are sent to the central server, where they are aggregated into a global model. Only parameter vectors are exchanged between the clients and the server during communication rounds, while raw data are kept locally, which complies with privacy-preserving constraints. Various algorithms were proposed for aggregating the locally learned parameter vectors into a global model, including FedAvg [McM+17], which updates the shared global model as follows:

$$\theta^{(r+1,0)} - \theta^{(r,0)} = \sum_{i=1}^{M} \alpha_i \cdot \Delta_i^{(r)} = -\sum_{i=1}^{M} \alpha_i \cdot \eta \sum_{t=0}^{T_i - 1} \nabla_i(\theta_i^{(r,t)}) \tag{3.2}$$

where $\theta_i^{(r,t)}$ denotes the model of client i after the t-th local update in the r-th communication round. Also, η is the client learning rate, $\Delta_i^{(r)}$ corresponds to the weight update from i-th client, and ∇_i represents the stochastic gradient computed over a mini-batch of samples.

Besides challenges such as communication efficiency, where the number of communication rounds has to be minimized, the heterogeneity of the data distributions across an extremely large number of clients and the inconsistency of the local objectives, which are highly witnessed in the distributed sensing environments, including IoT applications, represent the fundamental bottleneck being dealt with in the literature.

3.2 Induced heterogeneity

One of the major issues in federated learning is the heterogeneity of the clients. Heterogeneity (or equivalently diversity) goes beyond FL and is defined, for example, in ecology by [LR95] as: "the complexity and/or variability of a system property in space and/or time", where *system property* can be anything that is of (in this context, ecological) interest, *complexity* refers to qualitative or categorical descriptors of this property, while *variability* refers to quantitative or numerical descriptors of the property. Sampling from a heterogeneous system, i.e., exhibiting variability in one of its properties, yields observations with many differences between them. The system is said to "diverge from the ground state of perfect conformity" [Nun20]. Figure 3.1 illustrates one instance of the Yule-Simpson effect, which is a kind of heterogeneity in a setting encompassing different clients. Authors in [Kai+19, §3.1] survey some common ways in which data tend to deviate

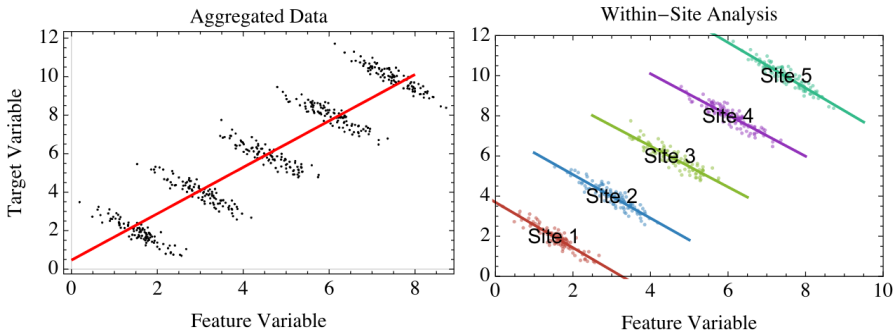


Figure 3.1: Illustration of the Yule-Simpson effect in a multi-center analysis setting. The correlations found in the individual sites versus when aggregating all the sites are reversed. Figure from [Nun20].

from being identically distributed. That is $P_i(x) \neq P_j(x)$ for different clients i and

j. By rewriting $P_i(x, y)$ as $P_i(y|x)P_i(x)$ and $P_i(x|y)P_i(y)$, this allowed authors to characterize different forms of induced heterogeneity including feature distribution skew (covariate shift), label distribution skew (prior probability shift), same label, different features (concept drift), same features, different label (concept shift), and quantity skew or unbalancedness. In the following, we illustrate these types of heterogeneity with concrete examples which are largely inspired by the IoT applications studied in this thesis and discussed in the introductory chapter 1.

The heterogeneity induced by the sensor's characteristics is, contrary to what could be thought, a major aspect of IoT applications, even if it is often dismissed in the literature. Indeed, even small variations, e.g., imperfections during the sensor manufacturing process, can lead to the introduction of impactful heterogeneous components [HO21b]. For example, in the context of HAR applications, authors in [KRM18] exhibited, in particular, one type of heterogeneity induced by the sensor characteristics, which is referred to as device-instance diversity where the signatures of the sensors exhibit variations, i.e., switching from one smartphone model to another (even the same model from the same constructor) while performing the same activity can lead to perceptible differences in the recorded patterns. These variations result, according to [Dey+14], from manufacturing flaws in the hardware that cause every sensor chip to react differently to the identical motion stimulus. Figure 3.2 illustrates the internal structure of MEMS accelerometer chip and the

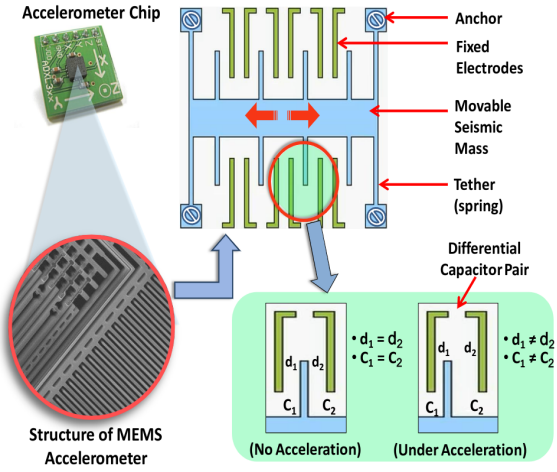


Figure 3.2: The internal structure of MEMS accelerometers and various components that are susceptible to suffering from imperfections (during manufacturing or deployment) lead to heterogeneity in the generated data. Figure from [Dey+14].

different components which are susceptible to generating imperfections leading to different fingerprints and, beyond the ability to track, can ultimately generate

heterogeneous components for the learning process. These sensor-, device- and workload-induced variations were investigated in [Sti+15] and have been shown to significantly impair the performances of activity recognition models. This phenomenon is often referred to, in the federated learning literature, as concept drift, according to [Kai+19], where the conditional distributions $P_i(x|y)$ may vary across clients (here exemplified by sensor, device, or workload) even if $P(y)$ (here, the target activities) is shared. The same label y can have very different features x for different clients [1]. Still, in the HAR applications, concept drift attracted more attention from the perspective of cross-user diversity where clients correspond, in this case, to the users who are known to exhibit important variations in the way they perform a given daily-life activity like running or walking. This is shown, for example, in [KRM18], to impair activity recognition systems based on a unique classification model approx. 30% classification accuracy drop.

Due to the pervasiveness of sensors that can be deployed on a massive scale to monitor diverse phenomena, the heterogeneity induced by the relativity of the viewpoints constitutes one of the major bottlenecks in learning processes. As we discussed in Section 6.1.4, the relativity of viewpoints generates variability in terms of the feature distributions that are captured for a given phenomenon. Although beneficial, this diversity has a perceptible impact on the performances of the learning processes when naively flattening the data being collected by the deployments. For example, Figure 3.3 illustrates the resulting linear discriminant analysis of three human activities (standing, running, and walking) captured by two different accelerometer-enabled devices (smartwatch and smartphone) placed at different on-body positions (hand and hip). This is typically what is known as feature distribution skew (or covariate shift). In this case, the marginal distributions $P_i(x)$ (i.e., the features being captured by each viewpoint) may vary across clients, even if $P(y|x)$ is shared (here in the case of HAR, the characterization of the target activities remain unchanged across viewpoints). Moreover, the relativity of viewpoints does not solely induce variability in terms of the feature distributions being captured but also has an impact on the forms of the label distributions. For example, depending on the exact location of the sensors as well as their sensing capabilities, what is sensed by a given client (sensor) may cover only partially (or even not at all) the values corresponding to a given concept (or target activity in the case of HAR applications) [2]. This is known as label distribution skew (or prior

[1] Note that the sensor characteristics do not induce only concept drift but also what is referred to as quantity skew or unbalancedness where the quantities of data that various clients can store vary greatly. This can be due, for example, to the diversity in terms of the sampling frequency across clients and also in terms of the energy and computational constraints (see § 1.2).

[2] Simiarly, in the case of entanglement between the cyber and physical domains, the sensors located on particular body locations may not capture (or not exactly) the right phenomena like temperature rise due for example to an infection.

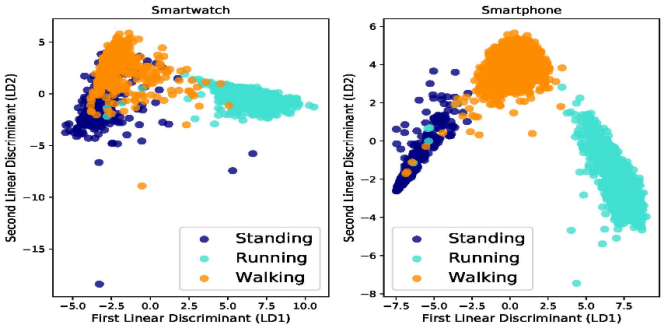


Figure 3.3: Linear discriminant analysis (LDA) of three human activities (standing, running, and walking) captured by two different accelerometer-enabled devices (smartwatch and smartphone) placed at different on-body positions (hand and hip). Considered features include: mean, standard deviation, and variance. Figure from [KRM18].

probability shift), where the marginal distributions $P_i(y)$ (i.e., the labels being captured by each viewpoint) may vary across clients, even if $P(x|y)$ is the same.

3.3 Datasets for heterogeneity study

In this section, we will briefly review the available datasets in the literature that are used to study the problems induced by heterogeneity in the federated learning setting—e.g., HHAR [Sti+15], "heterogenization" of existing datasets such as MNIST, etc. Datasets differ depending on the type of heterogeneity that is exhibited, and real-world FL datasets often include a combination of these [Kai+19]. Furthermore, there are obviously datasets that exhibit naturally-induced heterogeneity, mostly real-world datasets, while other datasets are constructed artificially to induce such heterogeneity into existing datasets in a controlled manner. This way, the proposed approaches can be assessed in terms of their robustness to different degrees of client heterogeneity. The idea here is to give a glance of the techniques used in the literature to construct such datasets and what are the goals pursued in each case.

On the one hand, synthetic non-IID datasets were, in many cases, trying to simulate label distribution skew, e.g. [McM+17; Hsi+20]. These non-IID datasets are formed by partitioning existing datasets like MNIST, CIFAR, or CINIC, based on their labels. Authors in [He+20, §B.3] provided a summary of the datasets and models used in the FL literature. Among the non-IID partition methods that have been listed in [He+20, Table 8], there are: "power-law", "realistic partition", "Pachinko allocation", and "latent Dirichlet allocation". On the other hand, many

examples of datasets exhibiting natural heterogeneity can be found in IoT applications, particularly in those we consider in the different experimental parts of this thesis. Beyond the SHL dataset [Gjo+18] for activity recognition which exhibits heterogeneity [3], other datasets in the literature have been specifically constructed for the study of the impact of heterogeneity induced by the sensing devices. These datasets are particularly interesting because of the heterogeneity induced by the imbalance of classes across clients and the heterogeneity induced by the perspectives provided by each position of the on-body sensor deployments.

Regarding the heterogeneity in activity recognition datasets specifically. As stated in Section 3.4, in activity recognition, the diversity of users, their specific ways of performing activities, and the varying characteristics of the sensing devices have a substantial impact on performances [Sti+15]. In these cases, the conditional distributions may vary across clients even if the label distribution is shared [Kai+19]. The Heterogeneity dataset for human activity recognition (HHAR) [Sti+15] was specifically constructed to investigate the impact of sensor heterogeneities on human activity recognition models. This dataset exhibits diversity in terms of the sensing modality, i.e., accelerometer and gyroscope, device type and manufacturer, i.e., smartwatches and smartphones, and workload, i.e., device CPU usage [4]. A total of 36 different sensors were tested on 9 different users in the context of activity recognition, e.g., 'Biking', 'Sitting', and 'Standing'. Furthermore, a large-scale aggregation of HAR datasets has been described in [Jan+17]. As the scale of the aggregated datasets is significant, the variation is induced by many different factors such as device type, acquisition protocol, users, sensor location, motion artifacts, and sampling rate. The comprehensive list of the aggregated datasets can be found in [Jan+17, Table 2].

3.4 Impact of heterogeneity on the FL setting

In this section, we provide an overview of the impact that heterogeneity across clients has on the FL setting, especially weight divergence and feature similarity in the learned models.

We saw above typical forms in which data tend to deviate from being identically distributed, e.g., covariate shift and concept drift. This leads to degradation in the performance of classical aggregation algorithms like FedAvg, which usually show strong empirical performance when confronted with IID data. This is due

[3] The description of the SHL dataset, which is used in the experimental evaluation parts of this thesis, is deferred to the next chapter.

[4] In smartphones, high CPU loads occur when a large number of applications are run simultaneously which makes the operating system prioritizes other tasks than those related to the sensing process. This may affect, in particular, the sampling process.

to objective inconsistency between the local empirical risk $f_i(\theta_i)$ and the global empirical risk $f(\theta)$ when the data are non-IID, i.e., $f(\theta^*) \neq \frac{1}{M}\sum_{i=1}^{M} f_i(\theta_i^*)$.

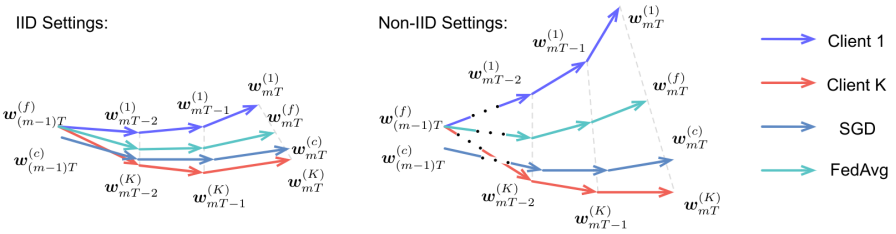


Figure 3.4: Illustration in the weight space of weight divergence in the federated learning setting. From [Zha+18]: weight divergence between clients (client 1 and K) and the central server (computed by SGD and FedAvg) in the IID and non-IID configurations.

In the weight space, the objective inconsistency manifests itself in the dissimilarity between the weights of each local model, more commonly called weight divergence in the federated learning literature. Figure 3.4 illustrates what weight divergence looks like in the weight space for non-IID settings. To better understand this phenomenon, various works have studied how weight divergence emerges across clients and, more precisely, across the layers of deep neural models. For example, Figure 3.5 shows the pairwise similarity of three different layers (the first layer, i.e., input, the middle layer (Layer 4), and the last layer, i.e., classification) across local models measured by the centered kernel alignment (CKA). Authors

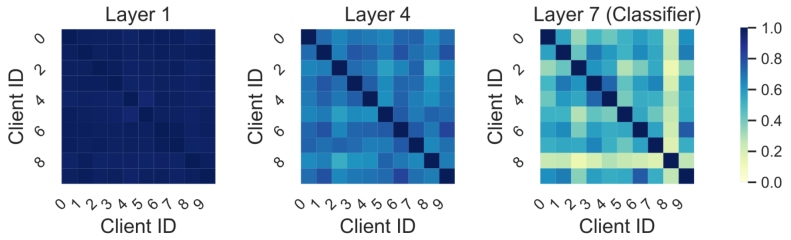


Figure 3.5: From [Luo+21]: pairwise similarity of three different layers (layers 1, 4, and 7) across locally learned models (clients 0—9). The centered kernel alignment is used to computed the pairwise similarity.

in [Luo+21] observed that the characteristics produced by the deeper layer have a lower CKA similarity than the ones produced by the upper layers. This suggests that the deeper layers of federated models trained on non-IID data have a greater degree of variability among various clients as compared to the upper levels. Moreover, to check the consistency of this result across clients, authors

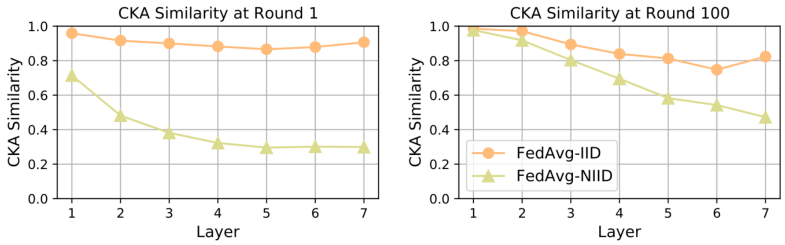


Figure 3.6: Figures from [Luo+21] showing a comparison of the CKA similarities as a function of the layer (1—7) in the IID and non-IID settings. CKA similarity at round 1 (left) and round 100 (right). The CKA similarities are obtained by averaging across different local models.

in [Luo+21] assess the evolution of the layer-wise average of the CKA similarities. The layer-wise average of the CKA similarities across clients allows for representing the feature similarity of a given layer across clients with a single value. These obtained values are depicted in Figure 3.6 and confirm what has been observed at the pairwise level, i.e., in comparison to the models that were trained using IID data, the models trained with non-IID data had invariably lower feature similarity across clients for all layers. Furthermore, weight divergence appears differently

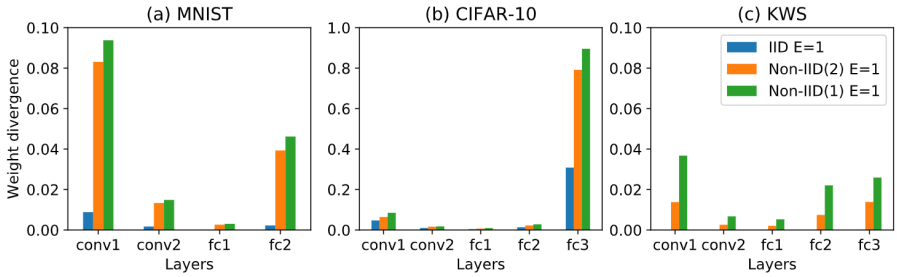


Figure 3.7: Figure from [Zha+18] illustrating the weight divergence of CNN layers (conv1, conv2, fc1, fc2, and fc3) in different IID and non-IID settings (IID, 2-class non-IID, and 1-class non-IID). Results obtained on three different datasets (MNIST, CIFAR-10, and KWS).

depending on the degree of heterogeneity. Figure 3.7 shows how weights diverge across layers in the case of a convolutional neural network trained on different datasets (MNIST, CIFAR-10, and KWS) with varying degrees of heterogeneity (IID, 2-class non-IID, and 1-class non-IID).

Overall, weight divergence is more important in non-IID training settings. Feature similarity across local models, measured by the CKA, decreases as we move toward deeper layers (classifier or output layer). Here, one can draw parallels with

meta-learning where the same phenomenon can be observed, i.e., task-agnostic and task-specific parameters tend to emerge in the same fashion in the frontend and backend layers of the learning models, respectively (see § 2.4). Interestingly, long lines of research try to identify and understand the mechanics of the learning process through which weight divergence emerges, to ultimately devise appropriate approaches, e.g., [Zha+18; Kar+20; TTN20; Red+20]. For example, Zhao et al. conducted in [Zha+18] a theoretical analysis on the root causes of weight divergence in non-IID settings. They proposed the following bounds on the weight divergence:

Proposition 3.4.1 (Bounds on the weight divergence [Zha+18, §3.1])**.** Given M clients, each with n_i IID samples following distribution P_i for client $i \in [M]$. If $\nabla_{\theta_i}\mathbb{E}_{x|y=j} \log f_i(x;\theta_i)$ is $\lambda_{x|y=j}$-Lipschitz for each class $j \in [Y]$ and the synchronization is conducted every T steps, then, we have the following inequality for the weight divergence after the r-th synchronization,

$$\left\|\theta_{(f)}^{(r+1,T)} - \theta_{(c)}^{(r+1,T)}\right\| \leq \sum_{i=1}^{M} \frac{n_i}{\sum_{i=1}^{M} n_i} (a^{(i)})^r \left\|\theta_{(f)}^{(r,T)} - \theta_{(c)}^{(r,T)}\right\| + \eta \sum_{i=1}^{M} \frac{n_i}{\sum_{i=1}^{M} n_i} \sum_{j=1}^{Y} \|P_i(y=j) - P(y=i)\| \sum_{t=1}^{T-1} (a^{(i)})^t g_{max}(\theta_{(c)}^{(rT-1-i)}),$$

where η is the learning rate, $g_{max}(\theta_i) = \max_{j=1}^{Y} \left\|\nabla_{\theta_i}\mathbb{E}_{x|y=j} \log f_i(x;\theta_i)\right\|$ and $a^{(i)} = 1 + \eta \sum_{j=1}^{Y} P_i(y=j)\lambda_{x|y=j}$.

A detailed proof of Proposition 3.4.1 can be found in [Zha+18, Appendix A.3.]. Although this result is specific to the extreme configuration where the data is sorted by class, and each client receives data partition from only a single class, it gives insights into the causes of weight divergence. Indeed, the idea behind Proposition 3.4.1, according to the authors, is that beyond the weight divergence at the previous synchronization step $(r-1)$, the probability distance for the data distribution on client i compared with the actual distribution for the whole population constitutes the two predominant components that cause weight divergence during the learning process. Furthermore, what is highlighted here is that clients' weight initialization is important, i.e., starting from different initializations exacerbates weight divergence, but it is dominated by the difference between the data distribution on client i and the population distribution when clients start from a unique initialization point. These kinds of results provide insights into how to deal with the impact that client heterogeneity incurs on the FL conciliation process. This is what we review in the next section.

3.5 FL approaches to mitigating heterogeneity

FedAvg sits on the idea of averaging the results of the progress made locally by clients in minimizing their respective objectives. As Konečnỳ et al. frame it: there is no reason to expect that, in general, the solution of 3.1 will be a weighted average of the local solutions unless the local functions are all the same—in which case we do not need a distributed algorithm in the first place and can instead solve the much simpler problem $\min_{\theta \in \Theta} f_1(\theta_1)$. Local objectives are necessarily heterogeneous and need more featured strategies to average them efficiently. Various approaches were proposed to handle non-IID (or heterogeneous) settings. These approaches can be categorized according to [Aca+20] into three main axes (or strategies): (i) modifying server-side updates; (ii) modifying device empirical loss dynamically; and (iii) using a decreasing learning rate. In the following, we will describe the principle behind these mitigation strategies and provide examples or approaches that implement these strategies.

The update of the server-side weights using vanilla FedAvg is performed via $\theta \leftarrow \theta - \Delta_i$, where $\Delta\theta = \sum_{i=1}^{M} \frac{n_i}{n} \Delta\theta_i$ (n_i is the number of examples, $\Delta\theta_i$ is the weight update from i-th client, and $n = \sum_{i=1}^{M} n_i$). This update process has been highlighted to induce convergence issues when confronted with client drift, for example [Kar+20]. Theoretical results, e.g.,[Li+19a], show that in non-IID settings, learning rate decay is critical for the convergence of FedAvg, and could lead to a solution at least $\Omega(\eta(E-1))$ away from the optimal after E SGD epochs if the learning rate is fixed. Controlling this update process in an adaptive and fine-grained fashion constitutes one way of tackling the heterogeneity of clients. Various methods have been proposed to modify this update rule. For example, FedAvgM (Federated Averaging with Server Momentum) [HQB19] propose server momentum, which is an adaptation of the well-known momentum method [Qia99], as a mitigation strategy to cope with clients heterogeneity. Indeed, momentum follows the same principle as in the physics of moving objects and works by keeping a running accumulation of past gradients in v, which is added to the update rule of SGD. This helps suppress oscillations during gradient descent and has been shown to have tremendous success in accelerating network training. In the context of FL, momentum can potentially prevent the server-side weights from diverging a lot because of largely dissimilar updates. The momentum is updated constantly with the new gradients as $v \leftarrow \beta v + \Delta\theta$, which is used then to update the server-side weights via $\theta^{r+1} \leftarrow \theta^r - v$. This seems particularly relevant for FL, where participating parties may have a sparse distribution of data and hold a limited subset of labels. Relatedly, the well-known adaptive optimizers in the deep learning community, including AdaGrad [DHS11], Adam [KB14], and Yogi [Zah+18], have been adapted in the federated learning setting by Reddi et al. in [Red+20]. The vanilla FedAvg update rule is augmented using adaptive step sizes, which are

used to adjust the learning rate (component-wise, i.e., a specific learning rate for each weight of the model). More precisely, a second-order moment estimate, v_r, of the past iterations is computed and added to scale up, i.e., high learning rates, or down, i.e., low learning rates, the updates to perform on the server-side weights. The model on the server is updated as $\theta^{r+1} = \theta^r + \eta \frac{\Delta_r}{\sqrt{v_r}+\tau}$ where v_r is computed following three different strategies: $v_r = v_r - 1 + \Delta_r^2$ (FedAdagrad); $v_r = v_r - 1 - (1-\beta_2)\Delta_r^2 \text{sign}(v_{r-1} - \Delta_r^2)$ (FedYogi); $v_r = \beta_2 v_{r-1} + (1-\beta_2)\Delta_r^2$ (FedAdam). Additional aspects around communication costs stemming from the necessity to maintain an additional state on the server are subject to trade-offs in these kinds of approaches. Furthermore, to make a parallel with the meta-learning strategies which specifically target adaptive learning rates (for each individual layer or module), discussed in § 2.4. Regarding this aspect, there are promising perspectives regarding bridging FL and GBML throughout adaptive learning rates, particularly for personalization aspects, because in GBML approaches, the notion of learning rate is tightly linked with the layers being either universal or specific to a task (or client).

Following our discussion on meta-learning and modularity (§ 2.4), increasingly more efforts are put further in the federated community towards the construction of fine-grained strategies for weight aggregation, similar to the ones presented in our works [OH22; HO22] and further detailed in Chapter 6. For example, authors in [Wan+20a] proposed the federated matched averaging (FedMA) algorithm, which uses a layer-wise strategy to construct the shared global model. In other words, element-wise averaging is often inefficient because of the permutation-invariance of the hidden neurons therefore, alignment of the neurons to match their counterparts across the clients before averaging can improve the conciliation step (see Figure 3.8). Note also that the work in [Wan+20a] makes use of the notion of permutation invariance of the neural network layers. We think that investigation could be carried out in the sense of the work in [RSP17] to devise fine-grained and more controlled strategies for weight aggregation. This aspect, in particular, is discussed in the conclusion chapter (Chapter 7).

Modifying device empirical loss dynamically is another means of handling client heterogeneity and is often achieved via regularization. Regularization is used in the machine learning literature as a way of reducing model complexity and ultimately getting improved generalization. Here, the regularization term serves as a penalty that pushes the parameters to converge to desired points of the parameter space and eventually prevents the weight divergence phenomena discussed in Section 3.4 and depicted in Figure 3.4. More precisely, the idea is to perturb the local function F_k in iteration t, as proposed in [Kon+16], by a quadratic term of the form $-(a_i^t)^T\theta + \frac{\mu}{2}\|\theta - \theta^t\|^2$ and make the nodes optimize for the perturbed problem instead. With this change, the improved method then takes the following

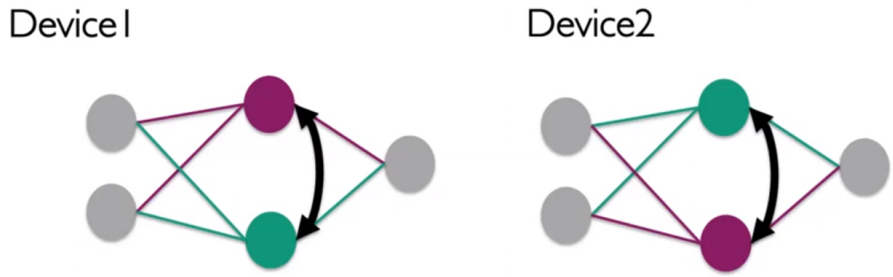


Figure 3.8: Figure from [Wan+20a] illustrates the permutation invariance of hidden neurons across devices (or clients), which requires an alignment phase before aggregation.

form:

$$\theta_i^{t+1} = \text{argmin}_{\theta\in\Theta}\, f_i(\theta) - (a_i^t)^T\theta + \frac{\mu}{2}\left\|\theta - \theta^t\right\|^2, \theta^{t+1} = \frac{1}{M}\sum_{i=1}^{M}\theta_i^{t+1}. \tag{3.3}$$

From an optimization point of view, as framed by Konečnỳ et al., the underlying idea here is to make each node (or client) $i \in [M]$ to "use as much curvature information stored in f_i as possible." Drawing inspiration from this form of iterations, various approaches were proposed in the FL literature [Li+20a; Kar+20; DBJ22]. For example, FedProx [Li+20a] consists of a dynamic regularizer, referred to as the proximal term, which is supplied by the server to effectively limit the impact of variable local updates. Far-from-server-model updates are penalized by this regularizer. Similarly, pFedMe [TTN20] also uses a proximal term, referred to as "reference point", which is also supplied by the server, and leverages Moreau envelope formulation of the modified local objectives to decouple the inner-loop optimization problem from the global model learning. Closely related, SCAFFOLD [Kar+20] tries to correct for this client-drift by estimating the update direction for the server model ($\mathbf{c}$) and the update direction for each client $\mathbf{c}_i$. Then, the difference $(\mathbf{c} - \mathbf{c}_i)$ is used as the estimator of the client drift, which is used to correct the local update steps [5]. The local models are, then, updated as $\theta_i^{(r+1,0)} - \theta_i^{(r,0)} = -\eta \cdot \big(g_i(\theta_i) + \mathbf{c} - \mathbf{c}_i\big)$. Figure 3.9 illustrates a simplified view of the update steps performed by SCAFFOLD in the weight (or parameter) space. Again, drawing parallels with the meta-learning approaches, regularization, as we discussed in Section 3.6, is similarly praised as a means to take advantage of

[5] SCAFFOLD is among the works that send extra device variables to the server along with the models. This leads to extra-communication costs that could be prohibitive depending on the transmission constraints imposed on the system. Approaches similar to SCAFFOLD like [Aca+20] try to account for the extra-communication costs when correcting for the client-drift.

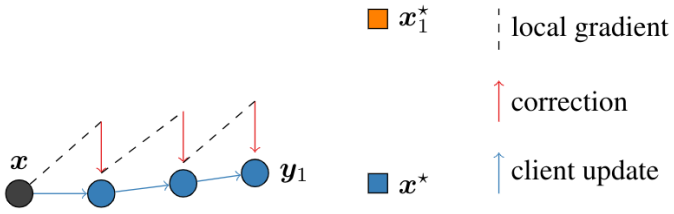


Figure 3.9: SCAFFOLD local updates. Figure from [Kar+20] illustrating the client drift (the local gradient represented by a black dashed line points to x_1^*—or θ_1^*) and the way it is rectified by the correction term $(\mathbf{c} - \mathbf{c}_i)$ making the client update point to the true optimum x^*—or θ^*.

task-relatedness (or task similarity) in the same way federated learning leverages regularization to mitigate heterogeneity. In FL, we mainly talk about mitigating heterogeneity of clients, while in metalearning, we rather have the goal of exploiting task-relatedness. While using different terminologies to describe strategies to tackle tasks and clients diversity, both FL and metalearning aim to leverage knowledge about either tasks or clients to improve the construction of a global and coherent theory across tasks and clients.

Many different approaches have been proposed to mitigate the impact of client heterogeneity in the FL setting. We started to notice in the course of the above sections that there are clear parallels between the federated learning setting and meta-learning based on the notions of task and client, e.g., the emergence of domain-agnostic and domain-specific parameters (or representations) § 2.4, tasks similarity versus clients heterogeneity § 2.6. This is particularly appealing to the IoT applications (or, more broadly, decentralized and distributed applications) as there are means to leverage both worlds to improve the learning process and the resulting models in terms of robustness, interpretability, etc.

3.6 Beyond tasks: families of (structurally) related clients

In the context of federated learning, as we have seen above, one of the major problems that the community faces is that of the heterogeneity of the clients (or data sources). This heterogeneity is linked to the diversity of clients who participate in the learning of a unified theory and manifests itself through different types of phenomena (listed in § 3.2). These are more or less exacerbated by the applicative framework of the Internet of Things considered in this thesis. Federated

learning approaches very often aim to overcome these phenomena by trying to identify the components that they induce within the data and circumscribe them. The ultimate goal is to manage to reconcile the data of the different clients and minimize their (apparent) discrepancies. In parallel, meta-learning approaches attempt to exploit the notion of similarity between tasks (or tasks-relatedness) in order to learn more general theories that are easily and quickly adaptable to new tasks. We saw in Section 2.6 that the notion of task translates in various ways in practice, e.g., dataset, user, learning environment, etc. Similarly, the notion of client can take different forms in practice, similar to those that tasks can take.

In many studies in the literature, this link between tasks (or domains), as considered in meta-learning, and clients in federated learning, has been investigated in a practical way. Authors in [Jia+19] leverage the connections between FL and meta-learning to build better personalized models for clients while authors in [CK21] studied convergence and accuracy trade-offs in local update-based approaches, including simultaneously the FL and meta-learning settings. For example, in [Smi+17], authors explored, among different practical problems, human activity recognition cast in a federated learning setting, where each individual user was considered to be a separate task allowing to benefit from the application of multi-task learning strategies. Indeed, the multi-task learning paradigm was considered in a distributed-compliant form [WKS16; WKS16] even though, as pointed out by, for example, authors in [Cal+18], in contrast with the standard federated learning settings, multi-task learning is usually explored in small data regimes (for adaptation) and a limited number of tasks.

More of a semantic distinction, which, nevertheless, has practical considerations: notice how the meta-learning community considers federated learning to be fundamentally a matter of personalization where the idea is to devise strategies to alleviate clients' heterogeneity, whereas meta-learning approaches look for commonalities between tasks and are, thus, fundamentally building upon the notion of task similarity. We think that this apparent duality deserves to be investigated further. This is one of the aspects that is discussed in our contributions, noticeably in Chapter 4, where the proposed problem formulation makes an explicit junction between these two aspects. Indeed, in meta-learning approaches, we seek to exploit the similarities between tasks in order to obtain better abstractions and better generalization capabilities. Apart from generalization capabilities, which are sought by all machine learning approaches, it is not exactly the same thing that is sought in federated learning approaches. Indeed, the latter approaches try to identify and isolate the components that induce heterogeneity. A form of communicating vessels is therefore possible—i.e., leveraging meta-learning approaches and multi-level modeling to improve FL learning. The idea, as we will show in the following chapters, is leveraging more featured structures and heterogeneity

isolation strategies to improve meta-learning approaches—and a fortiori beneficial for both domains. Beyond the few points of distinction, we, therefore, have a connection that appears between meta-learning and federated learning through the notions of tasks and clients. This connection is also the subject of numerous investigations in the literature [CK21; Jia+19; Kai+19; Kon+21; FMO20; Che+18; Smi+17].

A long line of research developed strategies to leverage meta-learning in the FL setting from the lenses of personalization—i.e., train an initial shared model and then adapt it to each client [Che+18; FMO20; KBT19; LYZ20]. In [Kai+19, §3.3.3] and [Kon+21, §V], authors enumerate works that leverage meta-learning to improve FL approaches in mitigating the impact of client heterogeneity, enable personalization, and few-shot learning. As we mentioned above, the idea is to extend meta-learning to FL by treating each device as a task. The goal then is to learn a global model shared across clients (just like the FL classical goal) with the additional property that it can easily be adapted by each client (hopefully, using only one or few steps of a gradient-based method) to ultimately match the locally generated data. The local models are said to be personalized and are consequently different from one another. The global objective of FL is reformulated in a way that the clients do not longer receive a final model (in the classical FL sense of the term) but a more appropriate bias (or prior), e.g., a good initialization, that puts the local learner in better dispositions to carry out learning. The bias learned globally is shared across clients and benefits, in a way, from the wisdom of crowds. For example, authors in [Che+18] proposed FedMeta, which implements this principle, i.e., a parameterized algorithm (or meta-learner) is shared instead of a global model.

This being said, as we saw, these existing works often tackle this problem primarily from the prism of "multi-model"-based approaches where multiple different models for different clients are used during inference in place of a single centralized model. Indeed, in [Kai+19, §3.1.1] authors, when discussing strategies for mitigating the impact of data non-IIDness on the learning process, make a legitimate, although arguable, case for customized models through local training, asking the question: "But if we have the capability to run training on the local data on each device (which is necessary for federated learning of a global model), is training a single global model even the right goal?". The idea is that given the computing capabilities that clients are dotted with, fitting individual models is becoming more feasible and appealing than having a single globally shared model. So, even if having a globally shared model may be required in certain scenarios, personalization turns the problem of heterogeneous data distributions from a bug to a feature, as Kairouz et al. put it. So, fitting individual personalized models for each local client is natural in this context.

Beyond personalization, the construction of a globally shared model in the FL setting has to take advantage of the similarities among clients (the dual of this being mitigating heterogeneity across clients). Indeed, as mentioned above (and discussed in detail in Section 2.6), one of the postulates of learning-to-learn is the existence of similarities between tasks that can be exploited in order to facilitate the learning process as well as the adaptation between tasks. In parallel to the heterogeneous aspects across clients, the similarities underlying the clients are ubiquitous, e.g., users may have similar behaviors, and sensing devices may capture the same aspects of the learning problems. Furthermore, very often in distributed sensing environments, knowledge about the relative geometry of the sensing devices and domain models describing the dynamics of the phenomena is available and can be leveraged and incorporated into the learning process. For example, the spatial structure of the sensors deployment and the induced views, sensors capabilities, and the perspectives (views) through which the data is collected (sensing model, range, coverage, position in space, position on the body, and type of captured modality) [AC09; WKA10; HO20]. For us, this is one of the facets of the problem that we are pursuing. Indeed, part of this thesis focuses on how to collaboratively learn a global model (or theory) instead of personalized local models. Noticeably, modeling the similarities among clients and integrating them via meta-learning strategies has promising potential for improving performance and boosting the effective sample size. For this, we take a closer look at how one can leverage the "meta-learning way" of constructing more general features and capturing local biases to improve, as a side-effect, FL approaches in mitigating heterogeneity.

Integrating these additional models describing similarities into the learning process has promising implications noticeably on the conciliation process of decentralized machine learning algorithms: one can exhibit the relative contribution of the individual views to the bigger picture. The primary goal is to develop a robust approach that integrates knowledge about the structure of sensing devices in a principled way to achieve better collaboration. The heterogeneity induced by various effects, in particular those related to distributed sensing environments, imposes some forms of mitigation. Noticeably, in Chapter 6, inspired by the fundamental principle of meta-learning, i.e., leveraging similarities among tasks, we model the structures underlying the data sources deployments using principled representation tools—e.g., special Euclidean group, which we use in order to capture domain symmetries and ultimately mitigate heterogeneity induced by varying point-of-views. Furthermore, as mentioned in Section 2.6, Chapter 5 leverages a form of task-relatedness by modeling the semantics of the label space, i.e., how atomic labels and groups of labels, to organize (or structure) the learning process. This form of task-relatedness leveraged in that chapter can be viewed from the perspective of clients, where grouping "semantically close" clients (according to

high-level criteria or objectives) to organize the learning process.

We saw in this chapter the challenges facing the federated learning paradigm and how these are dealt with in the literature. We paid special attention to the duality between the fundamental ideas of task similarity (in meta-learning) and client heterogeneity (in federated learning). This duality is detailed further in the next chapters, and structural constraints from the domain are leveraged to propose novel approaches.

Chapter 4

Integrating domain knowledge via structural constraints

In this chapter, we propose two novel approaches that leverage domain knowledge to select and augment learning examples. The main bottlenecks being dealt with in this chapter are the heterogeneity of the data sources and the cost of sensing and transmitting learning examples. The approaches presented in this chapter are based on the following works [OHB19; HO20; OHB21; HO23].

Besides naturally reducing the number of required learning examples needed to learn, providing appropriate learning examples has the potential to improve the performances of learning processes, i.e., alleviating heterogeneity impact and enhancing adaptability. In this chapter, we study the problem of providing appropriate inputs to guide the learner to reach better solutions in structured sensing environments and how it can benefit from available prior domain knowledge. Providing appropriate inputs is fundamental and has received long lines of research, and has been successfully used in various applications. Different strategies have been proposed ranging from curriculum learning [Ben+09] to self-paced learning [KPK10], and more recent works [HW19; Fan+18; Ren+18], where often the idea is to provide the learner with examples in increasing order of hardness, which is learned or determined heuristically. Various works have analyzed these strategies and tried to explain their performances by assuming them to be a particular form of continuation optimization method [AG12; Ben+09] or viewing it as a means for reducing the variance of the gradient estimator [Zha+19].

In this chapter, we propose a novel approach for selecting appropriate learning examples that leverages available domain knowledge. Precisely, our approach takes advantage of the availability of domain knowledge about the data sources, and the

way data are related to each other with particular structures. The sensing and transmission models of the data sources or their disposition in space and evolution through time are examples of domain knowledge that can be leveraged. These can be understood as privileged information in the sense of [VV09; VI15] and invariant predicates in the sense of structural risk minimization of [VI20]. By taking into account this additional knowledge, we constrain the space of curriculum that can be explored and has the advantage of accelerating the learning process, improving adaptation and robustness to unseen new situations, and enhancing data efficiency. In particular, we use invariants to encode the decision boundaries shared across the situations, and the learning examples sustaining these decision boundaries are used to form the curriculum. Portions of the decision boundary remain invariant to the evolution and heterogeneity of the sensing environments throughout the learning process and do not need to be adapted. The non-invariant portions, though, can be adapted using a handful of examples that support the decision boundary and which have to be identified. Using their distance to the decision boundaries, these examples ultimately form the curriculum. To some extent, this can be linked to continual/online learning approaches, e.g., [Alj+19; LR17], where the idea is to determine the examples to be saved in memory (replay buffer) and which may be sufficient in order to adapt the model to new tasks without forgetting the past, i.e., the current model does not degrade compared to old tasks, more commonly called catastrophic forgetting. Of course, in these contexts, the IID assumption might not hold anymore.

Specifically, the chapter is organized as follows. In Section 4.1, we formulate the curriculum selection as a problem where the evolutions of the sensing deployments are considered to generate new tasks that the learner has to adapt to. In Section 4.2, we devise a new approach that leverages domain knowledge in the form of invariants and decision boundary-supporting examples to select appropriate samples susceptible to guide the learning process to reach better solutions. This approach is empirically studied in Section 4.3 using different real-world industry 4.0 and IoT applications (Figure 4.4). We noticeably study the problem of adaptation to dynamic environments and heterogeneity of the data sources. Obtained promising results validate the benefits of the proposed approach in reducing the problem size and open-up perspectives regarding the incorporation of domain knowledge to improve learning performances in such environments. In Section 4.4, we describe the second approach based this time on the augmentation of learning examples informed by domain knowledge transformations. Again invariant aspects of the learned models are leveraged to guide the augmentation process. Ultimately, the universal portions of the learned models are intended to emerge and be reinforced, thus allowing them to be shared across federated clients, for example. The proposed approach is evaluated in Section 4.5 on an industrial

material engineering application introduced in [OHB21], which features various analytical models describing the chemical reactions involved in the material synthesis and used to constrain the augmentation. Section 4.6 discusses the proposed approaches in relation to the closely related literature and concludes this chapter.

4.1 Problem Formulation

We formalize this problem as a meta-learning problem [Sch87; TP98; FAL17], where the learner is constantly adapting its sample selection strategy throughout the span of the learning process by leveraging past experience and domain knowledge about the tasks it encounters.

4.1.1 Setting

In this chapter, we focus on learning in structured sensing environments, where a collection of sensors (also called data sources) are positioned at various locations of the space, on and around an object (concept or phenomenon) of interest, and generate streams of observations of a certain modality like acceleration, gravity, or video. We consider the standard setting of empirical risk minimization with parameters θ represented as a sum

$$\min_{\theta \in \mathbb{R}^d} \{ f(\theta) := \frac{1}{n} \sum_{i=1}^{n} f_i(\theta) \}, \tag{4.1}$$

where the function f_i denotes the empirical risks on the i^{th} subset of the training data, which often correspond to a few examples or to mini-batches. This objective corresponds also to the one optimized in the FL setting, where distributed clients collaborate to learn a unified theory. In this case, f_i is the local objective function at the i-th client, and the global objective is updated by aggregating the locally learned parameters. Similarly, this objective can be viewed in the context of an online stream of data as in continual learning applications [LR17; Alj+19; De +21] with no assumption about the distribution, such as IIDness. Therefore, depending on the application, the subsets of training data are not limited to mini-batches but can correspond to clients or segments of the data stream.

The ability to leverage past experience along with domain knowledge motivates framing our problem in a meta-learning setting. These approaches learn a meta-initialization $\phi \in \Theta$ for a class of parametrized functions $f(\theta)$ such that one or a few stochastic gradient steps on a few samples $\mathcal{D}_i^{\text{tr}}$ from a new task τ_i suffice to learn good task-specific model parameters $\theta_i = \mathcal{A}lg(\phi, \mathcal{D}_i^{\text{tr}})$, where $\mathcal{A}lg$ corresponds

for example to one or multiple steps of gradient descent [Raj+19]

$$\phi^*_{\mathrm{ML}} := \underset{\phi \in \Theta}{\operatorname{argmin}} f(\phi),$$
$$\text{where } f(\phi) = \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}(\mathcal{A}lg(\phi, \mathcal{D}_i^{\mathrm{tr}}), \mathcal{D}_i^{\mathrm{test}}). \tag{4.2}$$

This corresponds to a bi-level optimization problem since $\mathcal{A}lg(\phi, \mathcal{D}_i^{\mathrm{tr}})$ solves an underlying optimization problem. At meta-test (deployment) time, when presented with a dataset $\mathcal{D}_j^{\mathrm{tr}}$ corresponding to a new task $\tau_j \in \rho(\tau)$, we can achieve good generalization performance (i.e., low test error) by using the adaptation procedure with the meta-learned parameters as $\theta_j = \mathcal{A}lg(\phi^*_{\mathrm{ML}}, \mathcal{D}_j^{\mathrm{tr}})$. Here, instead of optimizing solely for a meta-initialization (biased towards fast adaptation to new unseen tasks), we propose to meta-learn a minimal subset of learning examples that can both alleviate the heterogeneity impact of the data sources and enhance adaptability to the evolution of deployments.

4.1.2 Tasks and task-relatedness

We have seen above that the notion of task translates into standard formulations in various ways in practice, e.g., datasets, mini-batches, users, clients, or data segments. Here, the notion of task is extended to embody data sources and portions (or segments) of the data streams being generated by these data sources.

Since the variations across the data sources can span, as we mentioned above, their disposition in space, their data generating processes, and their sensing models, every individual data source in the structured sensing environments that we consider can be considered to be a different task. Some studies [Smi+17; WKS16] investigated the link between tasks, as considered in meta-learning, and clients in FL, in a practical way. For example, [Smi+17] explored human activity recognition cast in FL setting, where each user was considered as a separate task. More generally, considering each individual data source as a separate task allows FL approaches to benefit from the application of multi-task learning strategies. Indeed, the multi-task learning paradigm was considered in a distributed-compliant form [WKS16] even though, contrary to the standard FL settings, multi-task learning is usually explored in small data regimes (for adaptation) and a limited number of tasks [Cal+18].

For dynamic environments, we assume that the evolutions that sensing deployments undergo lead to the emergence of new tasks. Works from the online learning community, e.g., [KBT19], consider meta-learning as the online learning of a sequence of losses, each corresponding to a different task. While having practical functional forms so that one can leverage the existing online convex optimization

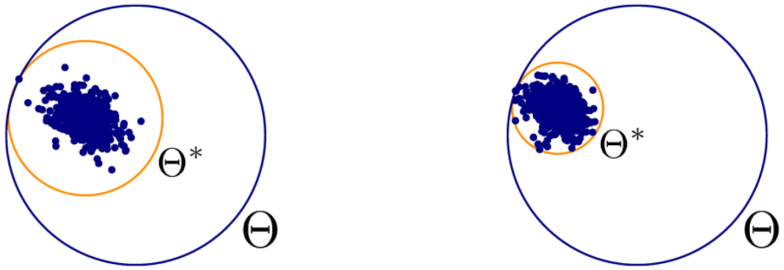


Figure 4.1: Parameter space after 1-shot adaptation (left) and many-shots (right) [BKT19].

literature, it says nothing about how data streams are partitioned into tasks, as it is done in reinforcement learning works, e.g., [Nag+19]. Indeed, since the evolution of the sensing deployments can occur at any time, new tasks can be formed at any timestep, e.g., in the case of industrial monitoring applications, the turbochargers' support which can sag over time or the natural wear of its seals are signs of deployment's evolution.

We, therefore, place ourselves in a scheme where the learner is confronted with a sequence of tasks $(\tau_1, \tau_2, \dots)$ which can actually correspond to different data sources or to portions of the data streams that these sources generate. Tasks corresponding to portions of a given data stream are formed whenever a distribution shift is detected and is assumed to be consistent during the entire segment. According to this formulation, tasks corresponding to different data sources (or sensors) arrive in a sequential manner in the form of a stream.

Task-relatedness. The need for tasks to be sampled according to a certain distribution $\rho(\tau)$ is a key assumption in the learning-to-learn setting [Bax00; GL21]. Indeed, there must be some link between observed activities and future unobserved tasks for meta-learning to occur. The fundamental notion used to make such an analysis and characterize how tasks are related structurally is the measure of task-similarity (or task-relatedness). Practically, existing approaches often measure task-similarity by computing similarity scores between tasks either via modeling their data-generating process or leveraging semantic information in the label space. In the case of gradient-based meta-learning approaches, where it is supposed the existence of a meta-parameter ϕ from which suitable task-specific parameters $\theta_i^* \in \Theta$ are reachable within a few steps, this is formalized via a small subset $\Theta^* \in \Theta$ where these task-specific parameters are supposed to lie (see Figure 4.1). Algorithms scaling with the diameter D^* of the subset Θ^* and with provable guarantees are developed, e.g., in [BKT19]. We assume that the distribution $\rho(\tau)$ shares some common structure. We suppose that tasks are related to each

other via transformations $g : \tau \to \tau$, assumed to control how the next task of the sequence is generated. We thus consider a sequence of tasks $(\tau_1, g_1, \tau_2, g_2, \dots)$ and the goal is to leverage these transformation to constrain sample selection. While scaling with the diameter of the subset of the parameters remains something we pursue, transformation-based relatedness is probably a more precise notion of relatedness than closeness in the parameter space.

4.2 Meta-Supervision via Sample Selection

In this section, we first provide a description of the main issues faced, noticeably the adaptability in dynamic sensing environments and heterogeneity of the data sources. Then, we motivate the necessity of providing appropriate curriculums in these kinds of environments, where additional domain knowledge about the structure of the data sources is available.

Due to the pervasiveness of sensors that can be deployed on a massive scale to monitor diverse objects, the heterogeneity induced by the relativity of the viewpoints constitutes one of the major bottlenecks in learning processes. The relativity of viewpoints generates variability in terms of the feature distributions that are captured for a given object. Although beneficial, this diversity has a perceptible impact on the performances of the learning processes when naively flattening the data being collected by the deployments. This is typically what is known as feature distribution skew (or covariate shift). In this case, the marginal distributions, i.e., the features being captured by each viewpoint, may vary across clients. Similarly, the dynamic nature of the sensing environments constitutes a major issue facing the learners, which have to continuously adapt to the evolution of, e.g., the transmission and sensing models of the data sources. Here, dynamic sensing environments can be understood as an analogous form of streaming setting where few examples or batches are received at a time. In this matter, the evolution in dynamic environments can be regarded as a form of heterogeneity but in the temporal dimension. Furthermore, reducing the number of learning examples is often a requirement because of the expensive cost to construct them, e.g., material engineering and chemical experiments, as well as the transmission constraints in distributed settings, e.g., large-scale IoT deployments.

In this work, we consider the problem of providing base learners in structured sensing environments with appropriate curriculums. Selecting relevant (or more appropriate) examples to present to the learner in each situation has the potential to improve learning performances (alleviating heterogeneity impact and enhancing adaptability) and naturally reduce the number of examples needed to learn. To do this, we leverage domain knowledge that is very often at the learner's disposal in such environments. This additional knowledge describes the data sources (taken

in its larger sense, i.e., datasets, mini-batches, users, clients, or data segments) in terms of their properties. In the simplest case, a property can be an integer corresponding to the index of a state space partitioning. More featured properties may include the sensing and transmission models of the data sources and relevant principles such as temporal coherence [Gol85], i.e., task-relevant properties of the world change gradually over time or proportionality of change to the magnitude of applied actions. These properties can be understood in the sense of privileged information, e.g., [VV09; VI15] and [LR17] in an online continual learning setting and could be leveraged to construct more appropriate curriculums.

At an abstract level, the idea of curriculum learning is that instead of presenting all examples at once, the learner gets access to the learning examples in an appropriate order so as to guide the learning process towards better solutions. The order of the examples is determined by their "easiness," and often, when the learning process starts, easier examples or those corresponding to the simplest concepts are favored. Concretely, curriculum learning may be divided into the following distinct but related problems: (i) sort the training examples by difficulty or complexity, referred to as *scoring function*; (ii) compute a series of mini-batches that exhibit an increasing level of difficulty, or *pacing function*, which defines the rate at which data is presented to the learner [HW19]. The main challenge is that often we are not provided with a readily computable measure of the easiness of samples [KPK10]. For example, [WCA18] sort the training examples based on the performance of a pre-trained network on a larger dataset, fine-tuned to the dataset at hand, while in [Ren+18; LH15], this is done based on an online approximation of their gradient directions.

4.2.1 Selection of learning examples constrained by domain-based transformations

We now specify the first variant of our proposed approach composed of two levels. As a running example, we illustrate the problem of adaptation on a real application, which corresponds to a sensing deployment used for the monitoring of a turbocompressor in real industrial conditions. Algorithm 2 summarizes the first variant of the proposed approach.

Algorithm 2: Selection of learning examples constrained by domain-based transformations.

Let $\mathcal{S}_v$ be the set of validated examples
Let $\mathcal{S}_n$ be the set of nominal examples of size ζ
Pick a first meta-initialization ϕ based on $\mathcal{S}_n$
for *task* $\tau \in [T]$ **do**
  Use the meta-initialization ϕ to perform predictions
  if $|x - y| < \delta + \epsilon$ **then**
    $\mathcal{S}_v \cup \{x\}$
  end
  Update the controller model based on $\mathcal{S}_v$ constrained by D^*'s rate of change
  Update the meta-initialization ϕ
  $\mathcal{S}_v \leftarrow \varnothing$
end

The algorithm starts by picking a first meta-initialization, $\phi \in \Theta$. When considering the turbocompressor monitoring application, this meta-initialization corresponds to the parameters learned during a nominal training period, where the industrial equipment is supposed to work in optimal conditions. The current version of the learner trained initially on a nominal period, ζ, is used to monitor the data streams generated by the data sources, and concurrently makes multi-step-ahead predictions.

The key step here is sample selection, where the predictions of the learner are used to validate real examples for the next generation of the controller model. As the task environment is assumed to be related via domain-based transformations, the validation of the examples for the next generation of the learner can be constrained by various forms of principles (e.g., temporal coherence and proportionality prior described previously). Whenever the discrepancy between the predicted output and the real system at a given time step is within the desired region of acceptable behavior, the corresponding data is used for the next generation. Previous steps generate a model of the system from data. During this process, the average limits between the generated signal from the model and the real system can be controlled using, for example, a standard defined by domain experts. Precisely, we ensure that $|x - y| < \delta + \epsilon$, where x (resp. y) are real examples (resp. model's predictions) corresponding to the current task. Similarly, abrupt changes caused by a substantial transformation, e.g., a bearing defect, can be constrained by the proportionality principle, i.e., the amount of change in task-relevant properties resulting from an action is proportional to the magnitude of the action. In terms of the diameter of the parameter space, these principles translate into a rate of change of D^* that is defined by domain knowledge and used to

constrain the task-specific parameters. After each task, the meta-initialization ϕ is updated with the validated examples.

4.2.2 Invariance and decision boundary

A learner's capacity for generalization is strongly affected by how effectively it learns the true decision boundary between the actual class distributions [CV95; Bis06]. In this second variant of our approach, explicitly enforcing invariants via the decision boundaries provides a powerful tool for selecting appropriate learning examples. The key intuition is that only a handful of examples are essential to support the decision boundary (see Figure 4.2).

Decision boundaries. The decision boundary of a learner f corresponds to its level set at zero and is denoted by $\mathcal{F} = \{x : f(x) = 0\}$, where x are learning examples. The boundary-supporting examples are defined as samples that lie near the decision boundary of the learner (see the bolded circles and square in Figure 4.2). They contain information about both the distance and the path direction from the base sample to the decision boundary. Authors in [MFF16] propose an algorithm to compute perturbations that fool deep networks efficiently. This method was used, e.g., in [Heo+19], in the context of knowledge distillation settings [HVD15] to provide a more accurate transfer of decision boundary information from teacher to student models. Here, we leverage this algorithm in order to project learning examples into the decision boundary. At an abstract level, given a learner that performs classification via the mapping $\hat{k}(x) = \text{argmax}_k f^k(x)$, where $f^k(x)$ is the output of $f(x)$ that corresponds to the kth class, projection is obtained via the minimal perturbation r that is sufficient to cross the level set and change the estimated label defined by [MFF16] as:

$$\begin{aligned} &\hat{k}(x) : \Delta(x; \hat{k}) := \min_r \|r\|^2 \\ &\text{subject to } \hat{k}(x + r) \neq \hat{k}(x), \end{aligned} \tag{4.3}$$

where $\hat{k}(x)$ is the estimated label. $\Delta(x; \hat{k})$ is usually referred to as the robustness of $\hat{k}$ at point x and is used to assess the robustness of classifiers to adversarial attacks, e.g.,[PC22]. In this variant of the proposed approach, we select samples by leveraging the boundary-supporting examples obtained using their corresponding minimal perturbations. We further exploit invariant portions of the decision boundary to refine the selected samples.

Invariance. In general, a mapping $h(\cdot)$ is invariant to a set of transformations G if when we apply any transformation induced by g to the input of h, the output

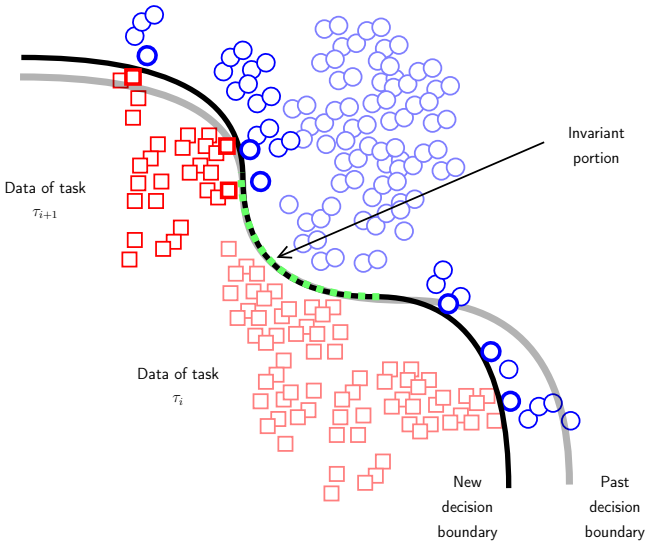


Figure 4.2: Invariant (or universal) portions of the decision boundary remain unchanged to data from new tasks. Only a few boundary-supporting examples are needed to adapt the non-invariant portions.

remains unchanged. A common example of invariance in deep learning is the translation invariance of convolutional layers. Formally, if $h : A \to A$, and G is a set of transformations acting on A, h is said to be invariant to G if $\forall a \in A$, $\forall g \in G$, $h(ga) = h(a)$. In the case of the SHL dataset, which features a structured on-body sensor deployment to monitor human activities, the elements g (belonging to the special euclidean group $SE(3)$) act on the spatial disposition of the data generators and, ultimately, the heterogeneity of the generated data: the variation of the data generated by a given data source should be proportional to the evolution of its location in the deployment. In dynamic environments, the generated data change in accordance with some given prior knowledge, and the idea is to enforce the learner to remain (or change) in accordance with prior knowledge.

This translates concretely into decision boundaries and boundary-supporting examples. Precisely, the portion of the decision boundary $\mathcal{F}_{[a;b]}$ bounded by points a and b is said to remain invariant to the action of the transformation g_{τ_i} (simply g_i) that takes τ_i and produces τ_{i+1}, i.e., $\mathcal{F}_{[a;b]}(\tau_i) = \mathcal{F}_{[a;b]}(g_i\tau_i) = \mathcal{F}_{[a;b]}(\tau_{i+1})$. The idea is to capture examples that support (or reinforce) invariant aspects according to a priori knowledge, in connection, to a certain extent, with the notion of k-priors from Khan and Swaroop and replay-based continual learning approaches, e.g.,[Alj+19], that can make the model adapt while keeping consistency with past episodes. The portion depicted in green in Figure 4.2 remains invariant to the ac-

tion of g_i, which acts on τ_i to generate τ_{i+1}, on the decision boundary. The other portions, however, change along with their supporting samples. Over time, according to the principle of meta-learning, universal portions of the decision boundary (which was learned so far) will remain as such (unchanged) throughout the span of the lifelong learning process. These universal portions can be thought of as the universal parameters (or meta-parameters) that are optimized for by gradient-based meta-learning approaches, where only a few (boundary-supporting) examples are needed to adapt the non-invariant portions.

Algorithm 3: Generic algorithm for learning to select learning examples. In the FL setting, the local models transmit the gradient corresponding to the selected learning examples and receive the updated server's model.

Pick a first meta-initialization ϕ
for *task* $\tau \in [T]$ **do**
  Project the examples into the decision boundary, which is based on the initialization ϕ (Eqn. 4.3).
  Determine the examples to use in order to update the parameters.
  Compute (exactly or approximately) the best fixed parameters θ^*_τ for task τ.
  Update ϕ and the portions invariant to evolution $\mathcal{F}^{\tau}_{\text{inv}}$ to be used for the next task (Eqn. 4.4).
end

Algorithm 3 summarizes the second variant of our proposed approach. It starts by picking a meta-initialization ϕ, similarly to the first variant. For example, to illustrate the projection on a simple case, when the learner is an affine function, $\Delta(x_0; f)$, is equal to the distance from x_0 to the separating affine hyperplane $\mathcal{F} = \{x : \mathbf{w}^\top x + b = 0\}$. The minimal perturbation to change the classifier's decision corresponds to the orthogonal projection of x_0 onto $\mathcal{F}$ (see Figure 4.3). This projection procedure can be instantiated with, for example, the iterative process proposed in [MFF16] or [Heo+19]. Even if it is not guaranteed to converge to the optimal perturbation, it was observed in practice that the algorithm yields good approximations of the minimal perturbations. Two important details are extracted from the projection of a new example x_0: (1) the distance $\Delta(x_0; f)$ between the new example and the decision boundary; and (2) the portion of the decision boundary on which the new example is projected. The selection of the examples is based on whether the portion of the decision boundary on which a given example is projected is invariant or not to the action of the group element g_τ. Optionally, the distance $\Delta(x_0; f)$ is used to rank the examples projected into

non-invariant portions of the decision boundary and ultimately select the examples that are closer to the decision boundary or impose an easy-to-hard order on the learning examples depending on their distance to the decision boundary, in the spirit of [Ben+09]. Using the selected learning examples, meta-update ϕ. The portions of the decision boundary invariant to the evolutions (heterogeneity) are updated as follows:

$$\mathcal{F}_{\text{inv}}^{t+1} = \mathcal{F}^{t+1} \otimes \mathcal{F}^{t} = \bigcup_{s<t+1} \mathcal{F}_{\text{eff}}^{s} \tag{4.4}$$

where $\otimes$ refers to an abstract operation on decision boundaries, which can be defined as in, e.g., [Heo+19], via *magnitude* and *angle* similarity measures between any two given decision boundaries. This could be a computationally impractical step. Here, we leverage the notion of effective decision boundary [LL93] which is defined as the intersection of the decision boundary and the regions where most of the data are located. This is a more appropriate process that matches the lifelong and distributed learning setting. Indeed, as we get access to the learning examples in an incremental fashion, the *effective* decision boundaries are stored as they get encountered during the learning process. In the case of FL settings, the model and the set of invariant portions are transmitted to the local learners. Note that in this same setting, instead of the learning examples themselves, it is the corresponding gradients that are transmitted to the central server.

4.3 Experiments

In this section, we empirically evaluate the effectiveness of the proposed approach on two real-world applications featuring structured sensing environments. In particular, the quantities of data needed to learn in each experimental configuration are assessed.

Experimental setup. We evaluate our proposed approach on two real-world benchmark datasets featuring structured sensing environments: the SHL dataset [Gjo+18] and the 102J dataset [OHB19].

The SHL dataset features a structured on-body smartphone-based sensors deployment capturing the motions of 3 users in their daily-life routines in the United Kingdom. It is a highly versatile annotated dataset dedicated to mobility-related human activity recognition. The dataset consists of motion sensor data recorded over a period of 7 months in 8 different modes of transportation in a real-life setting in the United Kingdom (*1:Still*, *2:Walk*, *3:Run*, *4:Bike*, *5:Car*, *6:Bus*, *7:Train*, and *8:Subway*). The dataset contains multi-modal data from a body-worn camera and from 4 smartphones, carried simultaneously at typical body locations (*Hand*,

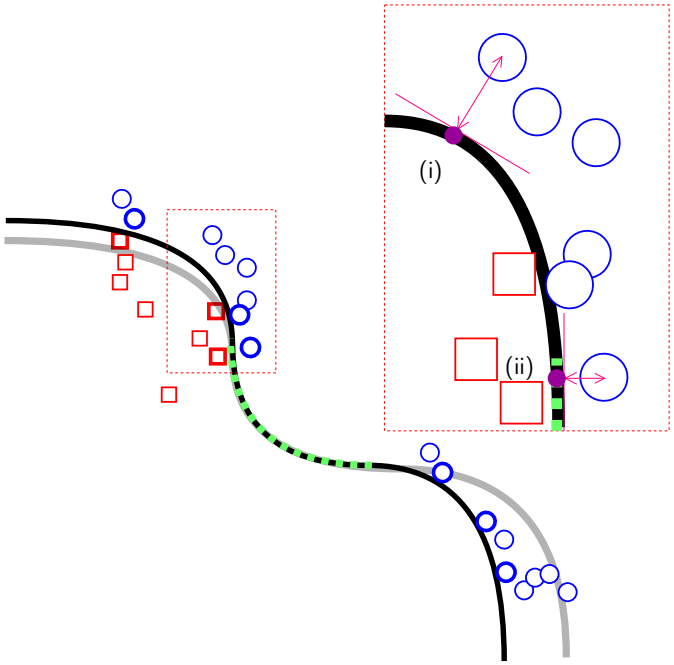


Figure 4.3: Projections (in purple) of the learning examples into the decision boundary in the case where the portion is (i) non-invariant and (ii) invariant to the action $g_\tau : \tau \to \tau$.

Torso, *Hips*, and *Bag*). The SHL dataset contains 3000 hours of labeled locomotion data in total, making it the most important in the literature. It includes 16 modalities such as accelerometer, gyroscope, magnetometer, linear acceleration, orientation, gravity, ambient pressure, cellular networks, etc.

Furthermore, we introduce a new real-world dataset (102Jdataset) featuring a structured sensing environment around a turbocompressor in real industrial conditions. Data were collected from a set of 10 sensors that continuously monitor a 102J turbocompressor operating in a real application (industrial conditions). The deployment topology exhibits, in particular, the location of the 8 vertical and horizontal vibration sensors as well as the 2 axial displacement sensors relative to the different components of the equipment, which allow us to assess defects that can appear in any of the three Cartesian directions. Acquisition of vibration data was carried out for each sensor at a sampling rate of 1 Hz, which is sensitive enough for capturing vibration trends.

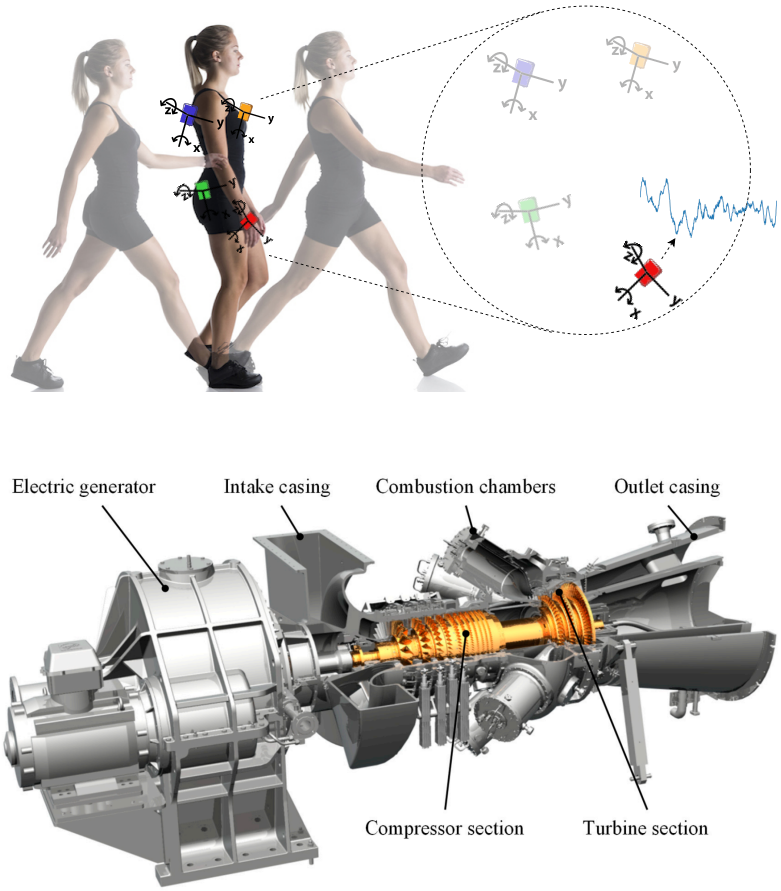


Figure 4.4: Structured sensor deployments in the SHL dataset for human activity recognition and the introduced dataset for turbo-compressor monitoring. Both applications took place in real-world conditions (see § 4.3 for a detailed description of these datasets).

4.3.1 Evaluation on human activity recognition

In the activity recognition application, the goal is to correctly classify the learning examples into their corresponding true human activities [1]. Evaluation of the learner's classification performance is based on the f1 score. We compare our approach with the following closely related baselines: DeepConvLSTM [OR16],

[1] In the HAR considered applications, activity recognition is addressed according to the following predefined chain [BBS14]: the labeled examples generated from the sensors are (1) segmented into short sequences; which are (2) pre-processed; and (3) from which discriminative features are extracted; (4) before being fed into a machine learning algorithm responsible of finding the mapping towards the activities (concepts).

Model	SHL F1 score (%)
DeepConvLSTM	65.3 ±.0206
DeepSense	66.5 ±.006
AttnSense	68.4 ±.03
Proportionality	73.6 ±.343
Boundary-supporting	75.3 ±.132

Table 4.1: Activity recognition performances (f1 score) of the baseline models on the SHL dataset.

DeepSense [Yao+17], and AttnSense [Ma+19].

- **DeepConvLSTM** [OR16]: a state-of-the-art HAR model encompassing 4 convolutional layers responsible of extracting features from the sensory inputs and 2 long short-term memory (LSTM) cells used to capture their temporal dependence.
- **DeepSense** [Yao+17]: a variant of the DeepConvLSTM model combining convolutional and a Gated Recurrent Units (GRU) in place of the LSTM cells.
- **AttnSense** [Ma+19]: features an additional attention mechanism on top of the DeepSense model forcing it to capture the most prominent sensory inputs both in the space and time domains and focus on them to make the final predictions.

In this application, the heterogeneity aspects stem from various factors, e.g., the displacement of the smartphones from their pre-defined initial on-body location. We conduct extensive experiments to evaluate the performance of the proposed algorithm in the following two settings: (i) we use a proportionality principle stating that the heterogeneity induced by the displacement of a given smartphone from its pre-defined initial position should be proportional to that displacement; (ii) we use the strategy based on boundary-supporting examples.

Table 6.5 summarizes the obtained results and shows that the proposed approach, in its two declinations, exhibits superior performance (5-7% improvement) compared to the closely related baselines. Figure 4.5 shows the experimental results corresponding to the heterogeneity induced by the smartphone displacement from their pre-defined initial locations. The smartphone displacement ($[5-40\%]$) is estimated w.r.t. the body segment the smartphone was located on initially, e.g., wrist-worn smartphone w.r.t. the entire arm. Both Full and BS are assessed in

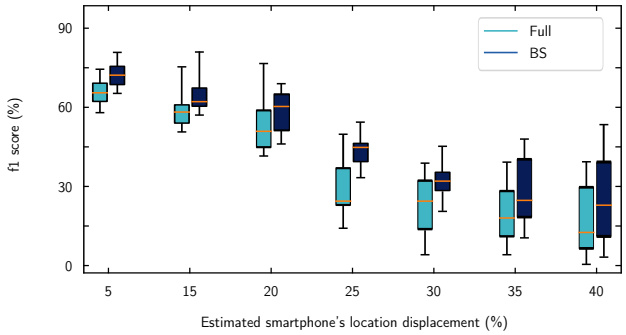


Figure 4.5: Activity recognition performances (f1 score) as a function of the estimated smartphone's location displacement from its pre-defined initial location. Full: training with all available learning examples; BS: training with boundary-supporting examples.

this set of experiments. As expected, when encountering larger displacements, the performance of both Full and BS degenerates and becomes near 30% (f1 score) starting from a displacement of 30% for BS, while Full has a significant drop in f1 score reaching $\leq$ 15% for a displacement of 40%. The proposed method achieves satisfactory improvements in terms of f1-score over the baseline methods when encountering displacements up to 25%. In particular, our proposed approach, BS, improves recognition performances by nearly 15% over the baseline, Full, using approx. 60% of available learning examples.

4.3.2 Evaluation on turbocompressor monitoring

We are interested in predicting abnormal vibratory phenomena in gas turbines that are susceptible to accelerating the deterioration of the system's components. Specifically, we model the vibration phenomena using a neural network-based autoencoder which is presented first along with the continuous monitoring solution that we provide.

The way for the autoencoder to learn generalizable encoding and decoding is to ensure that the number of hidden units is sufficiently restricted. Variants of the original AE models that make use of sparsity, denoising, or contraction were proposed [Rif+11] and are a way to free them from the information bottleneck and use encodings that are not necessarily smaller than the input dimensions. We need our autoencoder to be sensitive enough to recreate the original observation but insensitive enough to the training data such that the model learns a generalizable encoding and decoding. In order to explore various latent space representations that are more suitable to our particular context, we impose regularization via the sparsity constraint [Ng11]. It imposes that the activation level of the hidden

Model	Quality of reconstructions	Quantity of examples
Full examples	0.189 ±.032	100%
Temporal coherence	0.091 ±.06	70%
Proportionality	0.079 ±.024	68.3%
Boundary-supporting	0.071 ±.075	63.3%

Table 4.2: Obtained quality of reconstruction along with the quantities of learning examples required for the different evaluated models. Quantities of examples are normalized by the total number of examples in the dataset given a fixed segmentation window.

units remains low most of the time. For this, the average activation level $\hat{\rho}_j$ of a given hidden unit j is computed over all training sequences. The goal is to enforce $\hat{\rho}_j$ to be as close as possible to a target sparsity probability ρ (the sparsity parameter which is defined to be close to 0). This is done via the minimization of the Kullback-Leibler (KL) divergence between these two probability distributions

$$\sum_{j=1}^{n_{hu}} KL(\hat{\rho}_j||\rho) = \sum_{j=1}^{n_{hu}} \rho \log \frac{\rho}{\hat{\rho}_j} + (1-\rho) \log \frac{1-\rho}{1-\hat{\rho}_j} \tag{4.5}$$

where n_{hu} is the number of hidden units in the LSTM layers. To achieve this, we will add an extra penalty term to our optimization objective controlled by a parameter λ, which imposes the sparsity constraint. The whole model is then trained to minimize both the discrepancy between the original signal and its reconstruction and the divergence between ρ and $\hat{\rho}_j$:

$$J_{sparse} = J + \lambda \sum_{j=1}^{n_{hu}} KL(\hat{\rho}_j||\rho) \tag{4.6}$$

In the case of turbocompressor monitoring, we assess (i) a baseline where the old portions are kept in memory, and the model is trained using all available examples (Full); (ii) the first variant with standard-based example validation; (iii) the second variant involving the selection of boundary-supporting examples (BS). Evaluation of the learner is based on the quality of its reconstructions measured by the mean squared error (MSE) between the true vibration signals and the reconstructions. The standard-based example validation can be enforced either using: (a) temporal coherence, which is based on the boundaries defined by the standard ISO 20816, e.g., *threshold limit* defined between 7.1 and 18 (mm/s) and *not allowed red limit* up to 18 (mm/s); (b) the proportionality principle, which is

defined for each type of defect that occurs to the turbocompressor, e.g., the rate of change in natural vibration frequencies becomes observable when the crack depth ratio becomes greater than 0.30 [Tla+12]. Table 4.2 summarizes the obtained results in terms of the quality of reconstruction and the quantities of learning examples needed to attain reconstruction performances comparable to training with full examples. All the three proposed strategies achieve promising improvements in terms of reconstruction quality (9-11$\times 10^{-2}$ improvement) over the Full strategies. Boundary-supporting shows the most important improvements among the proposed strategies both in terms of the quality of reconstructions and quantities of needed examples. This suggests that, compared to enforcing domain-based transformations, leveraging examples located near the invariant portions of the decision boundary leads effectively to reinforcing these portions and not others.

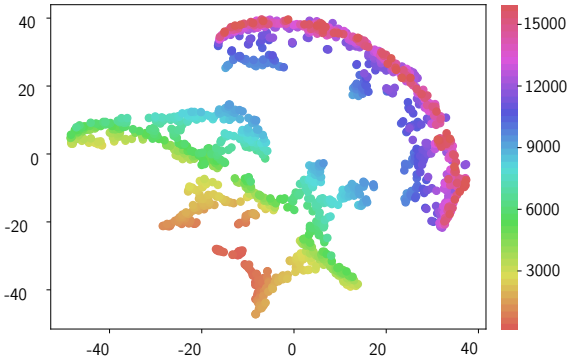


Figure 4.6: Reconstruction results showing the projection of the high dimensional latent representations of the autoencoder-based learner to a colored two-dimensional space using t-SNE [MH08]. Gradually-similar colors correspond to the sequential order of the hidden representations generated during contiguous periods of time.

Figure 4.6 shows the latent representations in a two-dimensional space obtained via the t-SNE algorithm [MH08]. The time-sequential order of the windows is depicted as the gradual variation of the color space where for example, purple corresponds to more recent windows in the monitoring process. Two distinct regions characterize the resulting latent representation space and correspond to a shift of the data distribution. It shows that contiguous signal sequences are projected to a continuous region in the latent space. This is a sign that our model's outputs evolve and adapt to the nominal evolution of the monitored system.

We also study the effect of an important part of our proposed approach, the nominal training period ζ being $\{200, 500, 1000, 1500, 2000\}$ w.r.t. the ability of the learner to not drift over time. Table 4.3 summarizes obtained results. Indeed,

Model@ζ	MSE	MAE
AE@200	0.1887	0.4259
AE@500	0.0715	0.2006
AE@1000	0.1657	0.3792
AE@1500	0.1829	0.4146
AE@2000	0.0375	0.1498

Table 4.3: Summary of the monitoring performances for various nominal training periods ζ. AE corresponds to an autoencoder, MSE: mean squared error, and MAE: mean absolute error.

it is expected that for shorter nominal training periods, the reconstruction model will not be able to consolidate enough its reconstruction capabilities and thus is susceptible to degenerate rapidly in a free-running configuration (where we do not have a mechanism for validating examples) and drift over time which will potentially result in a substantial amount of alarms and model replacements. We notice that regardless of the size of the nominal training period, the reconstruction model is able to maintain a negligible discrepancy over time, measured by the mean square and absolute errors. The same observation can be made regarding the number of alarms triggered by the controller model and the number of times it is replaced.

Overall, the proposed approach performs well on the two studied applications both in terms of classification or reconstruction performances and in terms of the quantities of data needed to learn.

4.4 Meta-supervision via data augmentation

Augmenting training examples is an important use case motivated by several real situations in decentralized applications, where real examples generated from real sensing experiments in the example (or sensing) space are missing or expensive to produce and transmit. Indeed, as we move far away from the set of real experiments or increase the distance between them, reconstruction models based exclusively on real experiments tend to be very unstable. In these regions, real experiments alone are not sufficient to satisfactorily determine the values of the state variables. From the perspective of the learning model's decision boundaries, the frontiers delimiting classes obtained using examples generated from real (sensing) experiments are noisy and sensitive to the hazards that are likely to interfere with the course of these experiments. We will focus in this part on how to account for the instability that taints models learned in the presence of these issues. Furthermore, in these types

of situations, the learner often has access to theoretical or analytical models, which can be used to obtain theoretically limited approximations where real examples are lacking. These models range from the simplest to the most sophisticated. We will leverage the available theoretical models describing the phenomena, the data sources, as well as the deployments in order to generate examples in the regions of the state space which require it.

In the experimental part illustrating the proposed approach, we will focus on an application coming from the field of material engineering and the synthesis of materials in an industrial environment, where real experiments are expensive to carry out, but theoretical descriptions of the chemical components involved are available. In this real-world application, we instrument our approach with a set of kinetics models, which originate from domain knowledge and are described in Section 4.5. However, the model that we describe here makes it possible not to be restricted only to analytical models of chemical reactions. We can go further by exploiting, for example, sensor coverage models, models describing how phenomena propagate in space and time, etc. For example, in wireless sensor networks, very often, the sensing capabilities of the sensing devices are described by sensing models. These models (or abstractions) encompass many different elements, including the direction of the sensing range (directional or omnidirectional) and the shape of the sensing area (deterministic or probabilistic). Figure 4.7 illustrates the shape of the sensing area for different sensing models. Furthermore,

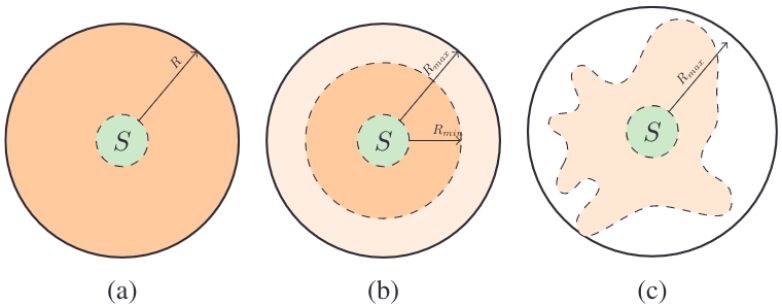


Figure 4.7: The shape of the sensing area for different sensing models: (a) Deterministic sensing model, (b) Elfes sensing model, and (c) shadow fading sensing model. Figure from [ESS19].

authors in [Mad+14] provide an example illustrating the combined modeling of a phenomenon and the sensing device used to capture it in the context of visual localization, mapping, and scene classification for autonomous road vehicles. In their work, authors model the spectral properties of the camera as well as those of the scene illumination and ultimately end up with an illumination-invariant color space that depends only on a single parameter derived from the image sensor specifications. A drastically reduced problem size. Figure 4.8 shows the interplay

between scene lighting (reflection geometry, surface reflectivity, material properties, etc.) and the sensor response (spectral sensitivity). These elements allow for deriving a good hypothesis space which only depends on the object's material properties.

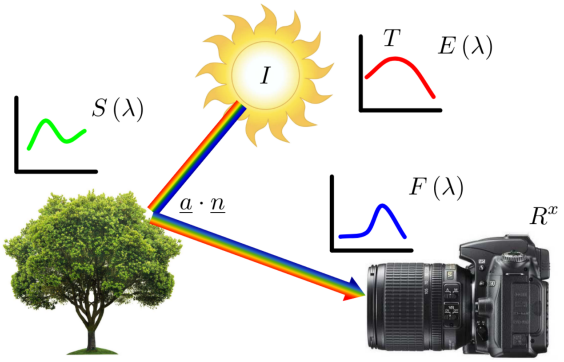


Figure 4.8: Illustration of the joint modeling process of the spectral properties of an image sensor (camera) as well as those of the scene illumination. Figure from [Mad+14]: shows three essential elements in scene processing, including (i) sunlight illumination, (ii) object of interest, and (iii) image sensor. Sunlight has intensity (denoted I) and spectral power distribution ($E(\lambda)$). The object of interest reflects light with an unknown surface reflectivity ($S(\lambda)$) and geometry term ($a \cdot n$), which depend on the material properties of the object and the relative angle of the light source and image sensor. The image sensor has a spectral sensitivity ($F(\lambda)$) and produces a response (R_x) at a particular location (x) on the imaging plane.

Data augmentation techniques consist in applying different operations with varying parameters on original data in order to generate (or synthesize) new data with certain properties. In vision applications such as image classification, the operations that can be applied to the original images include affine (or geometric) transformations (e.g., random horizontal flip, rotation, crop) and color space operations (e.g., color jittering) [Li+20b]. These techniques are widely applied in training deep neural networks and have been proven effective for improving the performance of learning models, noticeably, image classification models [Cub+19; Li+20b]. What makes data augmentation techniques work is that, as a result of increasing the diversity of the training data, the learning model is prevented from overfitting to the available original set of examples, especially when the quality of examples is poor and their quantity is not sufficient to generalize well, e.g., in the considered material engineering application or medical image analysis. Another reason behind the effectiveness of data augmentation techniques is their ability

to enforce (or impose) invariances that underly the true data distributions into the learning model directly. Indeed, in the case of face classification, for example, different images of a given face are often insensitive to illumination variations. Similarly, images of objects are often insensitive to horizontal flips or translation. Providing the learner with augmented images, e.g., using artificially generated illuminations or affine transformations, makes it more sensitive to the contents of the images rather than the variations. Traditionally, the way invariants are enforced into the learning models is by directly hardcoding them via network architectures: one such example is the convolutional neural network which encodes in its working the shift-invariance, for example. However, instead of explicitly hardcoding invariances into the model's design, it may be simpler to use data augmentation to integrate possible invariances [Cub+19]. In addition, the invariants may not be known and must therefore be learned, i.e., one cannot hardcode into the model architecture an underlying invariant that one does not know [2]

State-of-the-art in data augmentation dates back at least to the work of Beymer and Poggio [BP95], who built view-based and pose-invariant face recognizers by generating virtual views of faces from a single real view of a face. Their objective is to have the learned model capture the potential invariants underlying the distribution of face views. Since then, various works have been proposed, especially in computer vision [SK19], and organized into different categories depending on the way they perform augmentation, including geometric transformations, color space augmentations, generative adversarial networks, neural style transfer, and meta-learning. In some data augmentation approaches, the data manipulation operations are manually designed, while in others, this process is automated to better match the learning problem or dataset at hand. In the category of automatic image manipulation-based approaches, AutoAugment [Cub+19] has been the first to optimize the combination of augmentation functions through reinforcement learning. In these types of approaches, the sub-policies correspond to operations on the images, e.g., translate, rotate, auto-contrast, invert, and solarize, to which are associated hyperparameters controlling the probability that they are applied and the magnitude at which they should be. This is referred to as the search space. The choice of sub-policies is done automatically using, for example, a reinforcement learning process. Precisely, an RNN-based controller selects an augmentation policy from the search space. A child network is trained using the augmented data until convergence, achieving a performance that is used as a reward to update the controller so that it improves over time. Instead of relying on a discrete search problem, i.e., discrete selection of sub-policies, which is non-differentiable, improvements to this approach, such as Faster-AA [Hat+20] and DADA [Li+20b],

[2] Neural architecture search approaches correspond actually to learning architectures that encode specific biases [GH19].

suggest using a differentiable augmentation optimization strategy where the selection is smoothed to make it continuous. Furthermore, while AutoAugment and DADA, for example, do not account for any additional knowledge about the examples space in order to guide the selection of augmentation sub-policies, Faster-AA rests on the assumption that data augmentation is a process that fills missing data points of the original training data. As such, the controller's reward takes into account additional signals from the examples space.

4.4.1 Data augmentation based on domain transformations

Here, we propose a data augmentation approach guided explicitly by knowledge of the sample regions of space that require augmentation. It is formalized in the form of a bi-level optimization problem and exploits the capacity of this process to make invariant aspects emerge in the learned models. Figure 4.9 illustrates the proposed augmentation process informed by invariant aspects of the learned model.

The problem is formulated as a bi-level optimization process where a controller responsible for devising an augmentation strategy interacts with the learner so as to guide it to reach better solutions. Precisely, we propose a meta-learning-like interplay between the controller and the learner, where instead of learning an initialization for fast adaptation in downstream tasks, the controller learns to augment while guiding the learning process of the base learner. The motivation behind this formulation is similar to [Zho+21a] and consists of the promising ability of this kind of interplay to extract useful knowledge from related tasks. The process that we propose is roughly similar, in its spirit, to the GANs [Goo+14] approaches, where the idea is to try to deceive the learner (especially at the level of the boundaries between classes) by generating examples via an adversary which ultimately makes the learner much more robust. Here, we take a slightly different strategy: we make the controller generate examples related to invariant aspects of the learner, e.g., portions of the decision boundary or portions of the latent representations, which emerge and reinforce throughout along the interplay between the state (or example) space, the controller, and the learner.

State space partitionning

Here, we describe a generic representation of the example space illustrated using the real-world application that we consider in the experimental part (§ 4.5). To make it clearer when necessary, we also illustrate this generic representation with image-based transformations.

Let us consider a dataset $\mathcal{D}_{\text{real}} = \{\mathcal{X}^1, \ldots, \mathcal{X}^n\}$ consisting of n sets of experiments conducted at predefined conditions, e.g., specific location in sensor deploy-

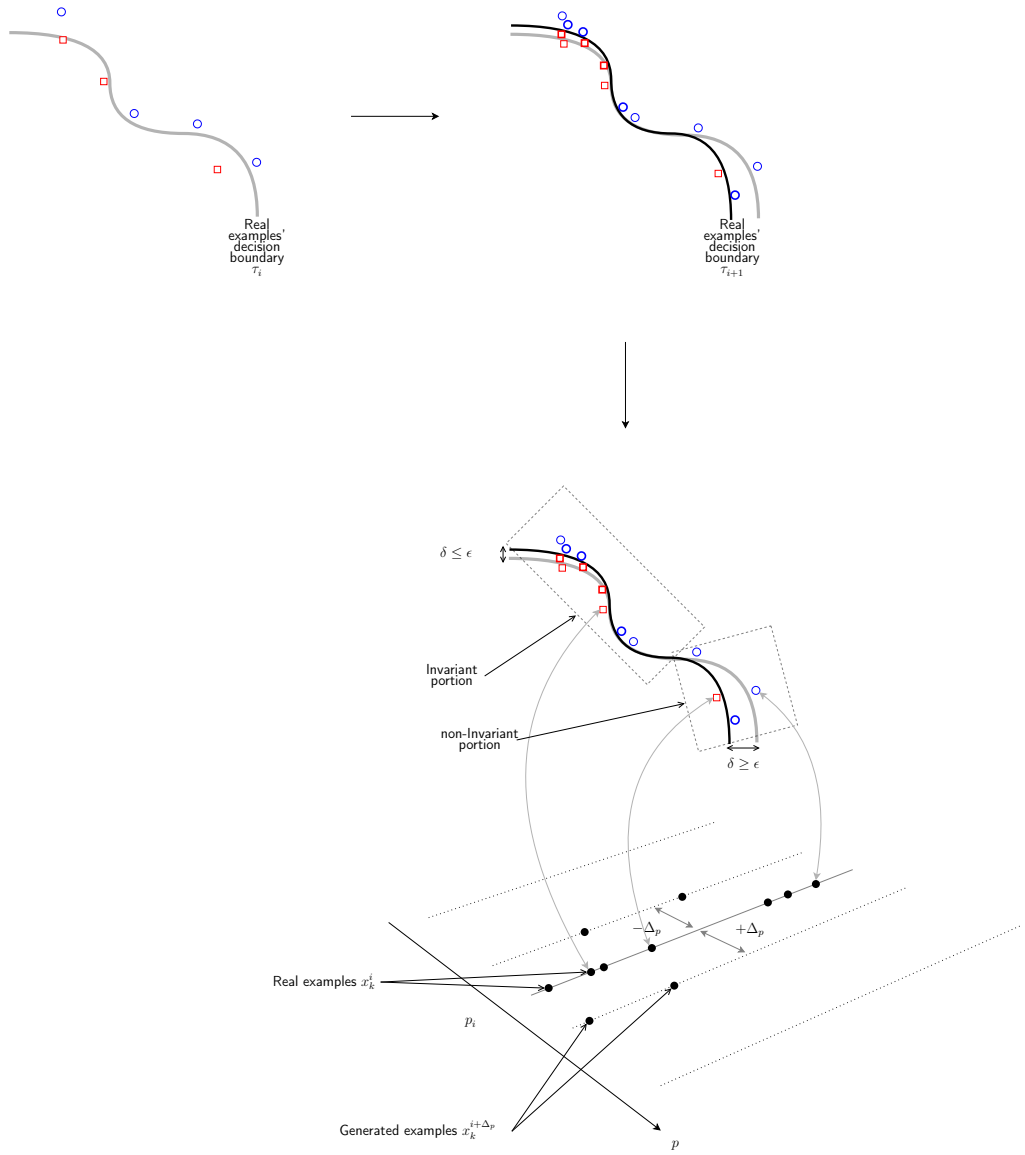


Figure 4.9: The framework of the proposed data augmentation approach. The portions of the decision boundaries that appear to be insensitive to domain transformations and that are reinforced throughout the learning process are used to guide the augmentation process. The idea is that domain transformations used to generate examples (e.g., image rotation) should translate into portions of the decision boundaries that remain invariant to these examples.

ment, lighting conditions or orientation of a face in an image, or ratio of a mixture of chemical reactants, and indexed by p_i, $i \in \{1, \ldots, n\}$. The difference between τ_i and τ_{i+1} corresponds, for example, to the percentage difference of a material; the distance between one client and another in a sensor deployment; the distance between two configurations of sensor characteristics; the difference from the point of view of their noise models. Figure 4.10 illustrates a representation of the state space and its subdivision into partitions. For example, in the case of the consid-

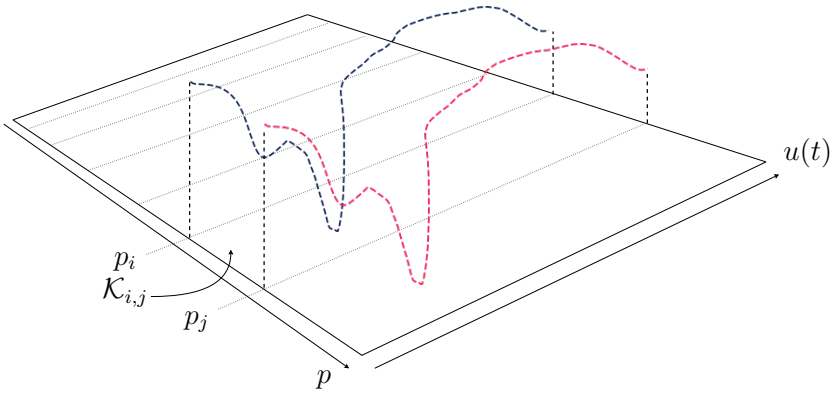


Figure 4.10: Representation of the state space and the subdivision into partitions $\mathcal{K}_{i,j}$, $i \in \{1, n-1\}$, delimited by the sets of real experiments (or examples) $\mathcal{X}^i$ and $\mathcal{X}^j$, in blue and red, respectively. In the considered application (TGA calcination process), these real experiments are conducted at p_i and p_j percentages of additional calamine oxide, respectively. Note that between any two sets of real experiments, i.e., inside each partition $\mathcal{K}_{i,j}$, no real experiment is available. We only have access to domain knowledge, e.g., in the form of transformations governed by analytical models, describing these partitions.

ered application (TGA calcination process), every single real example $x_j^i \in \mathcal{X}^i$, corresponds to the process of calcination applied to a given mixture of the considered chemical reactants (red pigment and calamine oxide) at a specific percentage p_i of the reactants and at a given temperature t_j. The result of an experiment is referred to as a state of the chemical reaction and is described (characterized) by what is referred to as state variables, e.g., heat flow, sample purge flow, the temperature of the mixture, and the mass of the mixture. Using the n sequences of real experiments $\mathcal{X}^i$ ($i \in \{1, n\}$), given in the dataset $\mathcal{D}$ described previously, ordered by the index of the predefined condition p_i ($i \in \{1, n\}$) at which the experiments were conducted, we divide the state space into $n-1$ contiguous partitions,

$\mathcal{K}_i$ with $i \in \{1, n-1\}$ such that $\forall i \in \{1, n-1\}$, $p_i < p_{i+1}$. These partitions are used as a search space for the controller from which it can sample augmented examples governed by transformations.

Outer-loop

The controller learns to generate examples that are intended to make the base learner capture invariant aspects in the data domain. The learning examples that can make the base learner capture the invariant aspects in the data domain should be crafted properly. In the literature, the controller often encompasses two important components: a search space and a search algorithm. The search space often consists of domain transformations that are applied to real examples in order to generate new ones. The search algorithm chooses, from the available search space, the most appropriate domain transformations to apply to the examples [3].

Here, we generate examples based on analytical models that theoretically approximate the actual values at particular locations in the example space. Examples are generated in their traditional form or as bounds. We assume that we have access to an example generator that can implement principles ranging from the simplest one, such as temporal coherence or proportionality prior (mentioned in more detail previously), to the most sophisticated one. In other words, we can have generators based on different analytical models. The generator is supposed to provide examples at the given position of the example space.

Illustration on the running example. In the case of the considered application, we leverage the kinetic models of the chemical reactions involved in the TGA calcination process. These models correspond to domain transformations that we apply to real experiments in order to generate new examples.

These kinetic models describe the time evolution of the mass as well as the temperature difference of the analyzed components during the thermal degradation process. The complexity of the chosen model depends on the desired objectives.

The simple method for obtaining kinetic parameters from experimental data is based on the kinetic equation $\frac{\partial \alpha}{\partial t} = k(1-\alpha)^n$, where $\frac{\partial \alpha}{\partial t}$ is the rate of the reaction (or decomposition). The constant k is given by

$$k = Ae^{-E_a/RT} \tag{4.7}$$

[3]For example, in AutoAugment, the search algorithm (implemented as a recurrent neural network) samples a data augmentation policy from the search space, which has information about what image processing operation to use, the probability of using the operation in each batch, and the magnitude of the operation.

where A is the pre-exponential factor, E_a is the activation energy and R is the gas constant. This is the Arrhenius equation [Lai84] which gives the dependence of the rate constant of a chemical reaction on the absolute temperature and the constants of the chemical reaction.

These kinetic models allow us to derive a series of penalty bounds $\mathbf{b}_j = [\Delta_j^{t_1}, \ldots, \Delta_j^{t_{max}}]$ at each applied temperature $t_1, \ldots, t_{max}$ using the neighboring points $p_i + \Delta p$, $p_i + 2\Delta p$, $p_i + 3\Delta p$, and so on (see Figures 4.11 and 4.12).

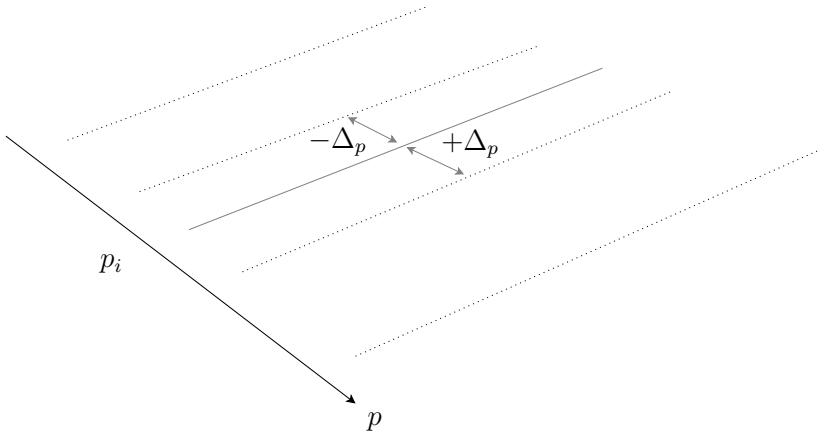


Figure 4.11: The example space is further subdivided using sets of points beyond and above each set of real experiments p_i separated by a step Δp, e.g., $\ldots$, $p_i - 2\Delta p$, $p_i - \Delta p$, $p_i + \Delta p$, $p_i + 2\Delta p$, $\ldots$ These sets of points do not correspond to real experiments. Instead, they correspond to areas where the example generator is utilized to generate examples based on domain knowledge.

Invariance to domain transformations. We are interested in the aspects that are invariant to the transformations g_i, acting on task τ_i to generate another task τ_{i+1}, and how to reinforce them throughout the learning process. Tasks here are exemplified by the sets of real experiments $\mathcal{X} \in \mathcal{D}_{\text{real}}$.

The key element that we introduce here is that the controller should augment only where it is necessary, i.e., where particular aspects of the base learner need it most. The reward returned to the controller must therefore guide it to choose where the examples should be generated. This is equivalent to guiding it to select the best sub-policy (or transformation) to apply to real examples [4]. Here, as we mainly

[4] In Faster-AA, the proposed approach attempts to generate examples in regions of the example space where real examples are not available. The goal is, in particular, to increase the diversity of the data distribution. No additional constraints on the regions to explore are imposed.

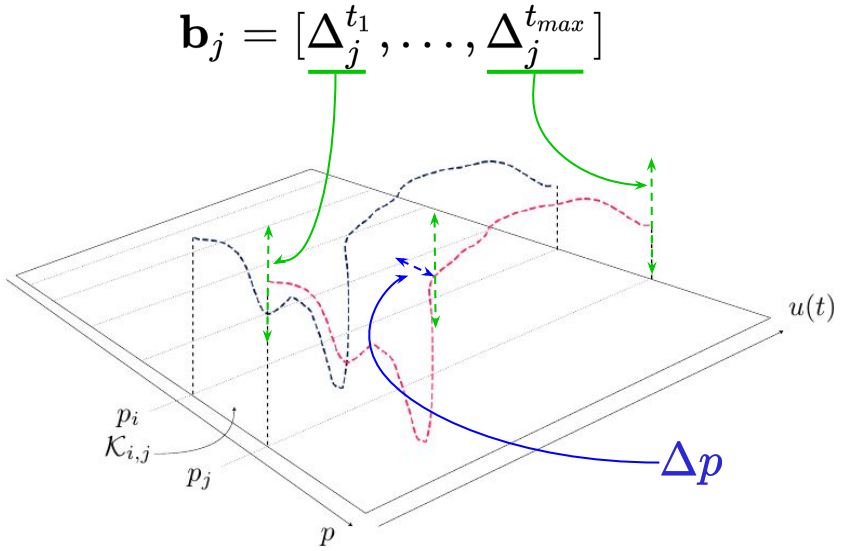


Figure 4.12: Given a set of real experiments (in red), using the neighboring points $p_i + \Delta p$, $p_i + 2\Delta p$, $p_i + 3\Delta p$ (in blue), we derive a series of penalty bounds $\mathbf{b}_j = [\Delta_j^{t_1}, \ldots, \Delta_j^{t_{max}}]$ (in green) at each applied temperature $\{t_1, \ldots, t_{max}\}$.

look for strategies that can guide the learner to capture invariants in the data domain, we seek controllers that learn to generate examples based on the invariant aspects of the learner's model. The idea is to craft augmented examples that have the ability to make these particular aspects emerge and reinforce throughout the learning process. Ultimately, the invariant aspects that will emerge using this process can be shared with other tasks: a kind of shareable universal components (see Chapter 6 where universal components of the data are described in more detail).

We would like to learn meta-parameters that can effectively capture the transformations underlying the different tasks. Specifically, the components of the model that are fixed are those related to domain knowledge (e.g., kinetic model) and whose evolution during the learning process is negligible. The other components are related either to the differences across tasks or to the noises tainting the experimental process during which the examples are generated. These components are likely to evolve. The meta-parameters are intended to capture the transformations g_i acting on the tasks, precisely, the aspects of the learner that are invariant to the action of these transformations, which we recall are unknown but simply supposed to govern the transitions from one task to another.

This objective can again be materialized (as in the previous approach § 4.2) by the portions of the decision boundaries which remain invariant to the actions of

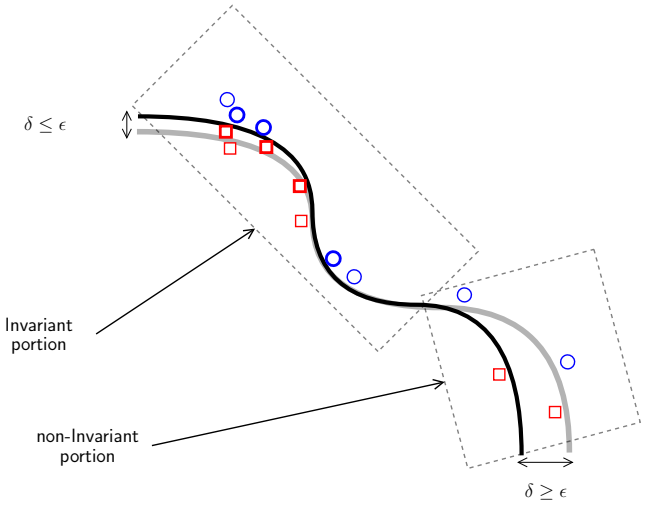


Figure 4.13: The idea is to reinforce the invariant portions of the decision boundary ($\delta \leq \epsilon$) in order to make it more robust, i.e., universal components of better quality shared across tasks (or sets of real experiments).

the transformations (see Figure 4.13). The process starts by picking a first meta-initialization ϕ based on one of the available sets of real examples, $\mathcal{X} \in \mathcal{D}_{\text{real}}$, corresponding to the end results of real experiments (or sensing processes). This results in an initial version of the model's decision boundary depicted in Figure 4.14 (gray color), which is very often impacted by the approximations and noise tainting the real experiments. The next step is to decide which regions of the example space (and a fortiori, the domain transformation) should be used to update the model. Faster-AA, for example, fills missing data points of the original data distribution by minimizing the Wasserstein distance between this distribution and that of the augmented examples [5]. Here, we use examples supporting the decision boundaries that remain invariant to the action of the transformation g_i (see Figure 4.14). The controller uses the set of invariant decision boundaries (as defined in the previous approach § 4.2) to find the corresponding domain transformation $j\Delta p$ that can be used to reinforce it.

[5]Precisely, the authors use adversarial training in order to minimize the distance between the distributions of the augmented data and the original data. The Wasserstein distance between these distributions computed via Wasserstein GAN [ACB17] makes the distribution of augmented data get closer to the distribution of the original data.

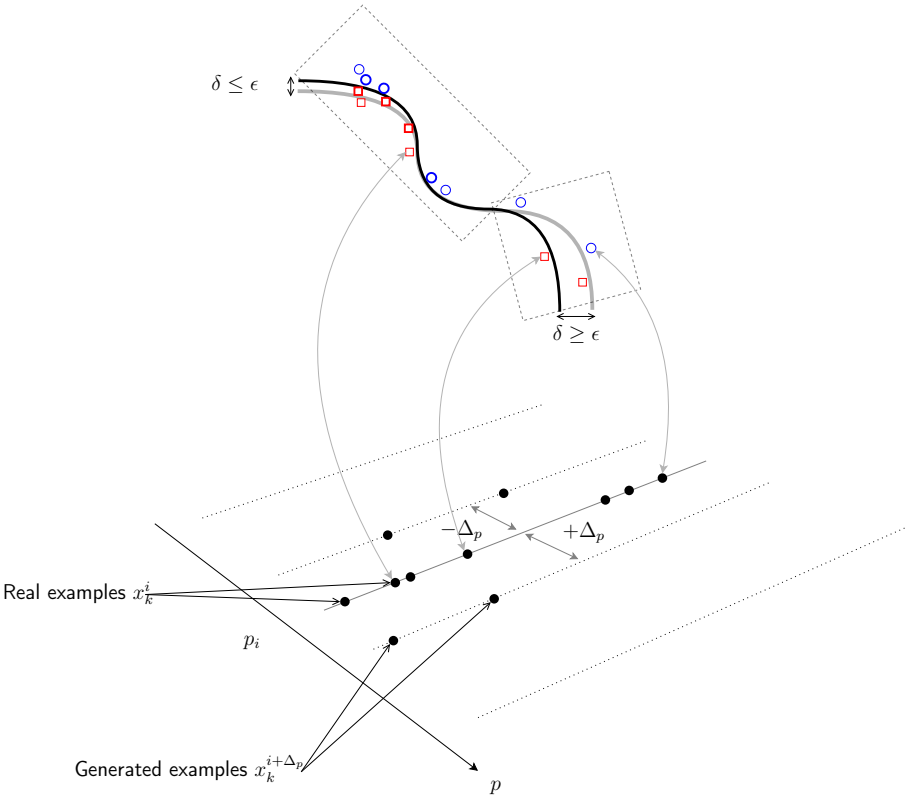


Figure 4.14: Illustration of the interplay between the model's decision boundaries and the example space. The model's decision boundaries are supported by examples from the example space. The controller learns to generate examples that make the base learner capture invariant aspects in the data domain. These invariant aspects are reinforced by generating examples in certain regions of the example space.

Inner-loop improved with Regularization

After generating the learning examples, the base learner fits its parameters to these examples. We define a partition where real experiments are conducted in regions greater than p_j but obtained by the extension of the model between two sets of real experiments $\mathcal{X}^i$ and $\mathcal{X}^j$. We denote this partition $\mathcal{K}_{i,j}$ $(i, j \in \{1, n\}$ and $i < j)$. Two kinds of models are considered: (1) a model that approximates experiments in the partition $\mathcal{K}_{i,j}$ circumscribed by two given sets of real experiments; (2) a model that approximates experiments for the partition beyond a fixed partition $\mathcal{K}_i$.

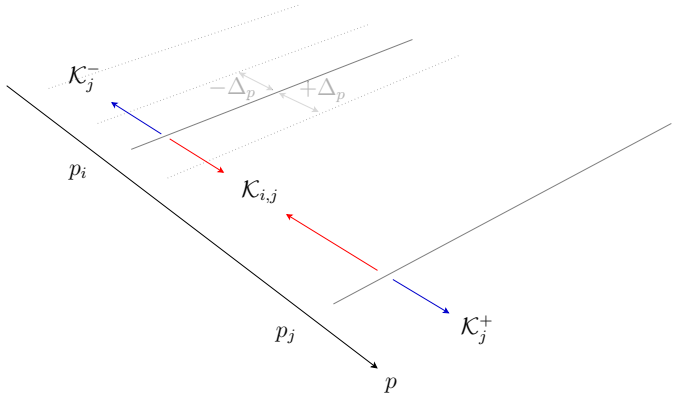


Figure 4.15: Representation of the state space partitioning with respect to the evaluation protocol devised to assess the extent of the model's predictions (or reconstructions). Recovering inside circumscribed regions $\mathcal{K}_{i,j}$ and Recovering outside a circumscribed region $\mathcal{K}_j$.

- **Recovering inside circumscribed regions.** In this configuration, we try to recover the partition $\mathcal{K}_{i,j}$ from its delimiting sets of real experiments $\mathcal{X}^i$ and $\mathcal{X}^j$. For this, we train a model $\theta_{\mathcal{K}_{i,j}}$ using all elements of these two sets, and we perform validation on the set of experiments $\mathcal{X}^k$, with $i \leq k \leq j$.
- **Recovering outside a circumscribed region.** In this configuration, we use sets of real experiments $\mathcal{X}^i$ in order to recover partitions $\mathcal{K}_k$ that fall outside the sets of experiments conducted with $\mathcal{X}^i$. In this case, we train a model $\theta^k_{\mathcal{K}_i}$ using all elements of partitions labeled by j such that $j <= i$, and we perform validation on the set of experiments $\mathcal{X}^k$ such that $k > i$. Here, we want to assess the ability of these models to extrapolate to other partitions and to what extent they can do that.

Here, we integrate the augmented examples provided by the controller into the learning process via a regularization-like process. The regularization process is known to reduce model complexity by penalizing the learner for choosing complex models and thus alleviating the model from overfitting the available data. Here, the idea of using this kind of process is to penalize "structures" that are not consistent with the examples (or bounds) generated by our controller. This is a weak form of supervision that has been shown effective in the literature [SE17] and is motivated by the simplicity it offers for expressing distinct properties of the models to learn, e.g., physical properties or domain models, instead of a direct input-output form of examples. This additional regularization-like term is derived from the augmented

examples and allows the trained models to stay in a theoretically bounded range. We derive from these models an additional term, $R : \Theta \rightarrow \mathbb{R}$, that is plugged into the original optimization objective, which, then, becomes:

$$\theta^* = \underset{\theta \in \Theta}{\operatorname{argmin}}\, \ell(\theta) + \lambda R(\theta) \tag{4.8}$$

where $\lambda \in]0; 1]$ is a weight parameter used to control the impact of the regularization term. This term is used precisely at the interfaces between the various sets of experiments $\mathcal{X}^i$ that partition the state space. By adding this regularization term to the standard empirical loss function, the model considers both the mean squared differences between model prediction and real experiments as well as the divergence from the governing kinetics models (as reflected by the second term in Equation 4.8).

Algorithm 4: Generic algorithm for learning to augment examples.

```
Pick a first meta-initialization φ
for task τ ∈ [T] do
    The controller samples an augmentation policy.
    for j ∈ {−p, ..., +p} do
        Generate examples based on the sampled
         augmentation policy (Eqn. 4.9).
        Compute (exactly or approximately) the best fixed
         parameters θ*_τ for task τ using augmented set of
         examples D^aug_train (Eqn. 4.8).
    end
    Get a reward in the form of validation accuracy
     (depending on the extent).
    Update the controller's augmentation policy using the
     obtained reward.
    Meta-update the meta-parameters φ.
end
```

Illustration on the running example. We leverage the pre-exponential factor and its variations for small increments Δp of the percentage of the two components in the mixture. Additional concentrations of some components imply more molecule collisions and, thus, an increase in the pre-exponential factor. We compute the pre-exponential factor numerically for these small variations and use them to encode the desire for the continuity of state variables values for variations of the mixture percentage.

At any given calamine percentage p_i, we compute numerically the kinetic constant k, which defines the kinetic energy of the reactants. This allows us to derive a series of penalty bounds

$$\mathbf{b}_j = [\Delta_j^{t_1}, \ldots, \Delta_j^{t_{max}}] \tag{4.9}$$

at each applied temperature $t_1, \ldots, t_{max}$ using the neighboring points $p_i + \Delta p$, $p_i + 2\Delta p$, $p_i + 3\Delta p$, and so on. The regularization-like term becomes

$$R(\theta) = \frac{1}{P} \sum_{j=1}^{P} \mathbb{1}\{|f_\theta(p_i + j\Delta p) - \mathbf{b}_j| > \epsilon\}, \tag{4.10}$$

where P is the number of neighboring points and depends on both the distance between the sets of experiments and the extent of the small increments Δp. This additional term provides a necessary constraint, which our model must satisfy. We thus push our model in the direction of better satisfying both terms of the cost function. At this level, the penalty bounds can be provided, for example, by the sensor coverage models (see Figure 4.7) or by the spectral properties of the camera and scene illumination (see Figure 4.8). In the experimental part, we focus on analytical models describing the dynamics of chemical transformations.

Finding quality solutions

One way to incorporate domain knowledge is through conditional stochastic gradient descent. It constitutes a surrogate which makes it possible to restrict the size of the space of the hypotheses. Indeed, a given set of parameters, i.e., a point in the parameter space as shown in Figure 4.16, is in itself an assumption. This means that the conditional gradient descent as a process implements a form of restriction on the valid hypotheses. See Figure 4.16.

Within the context of our application (and of the approximation capabilities that we are targeting), the regularization-like term that is glued to the cost function (Equation 4.8) is essential in order to force the model to take into account the continuum between the different p_i, and thus obtain optimal solutions. However, during the optimization process, convergence towards a solution satisfying both terms of the equation simultaneously is not ensured. The optimality criterion for us corresponds to finding so-called Pareto-optimal solutions such that none of $\ell(\theta)$ or $R(\theta)$ can be made better without making the other worse. Using the Langrangian interpretation, Equation 4.8 is the same as the following constrained formulation,

$$f^* = \underset{\theta \in \Theta}{\operatorname{argmin}}\, \ell(\theta) \text{ s.t. } R(\theta) \leq \mu, \tag{4.11}$$

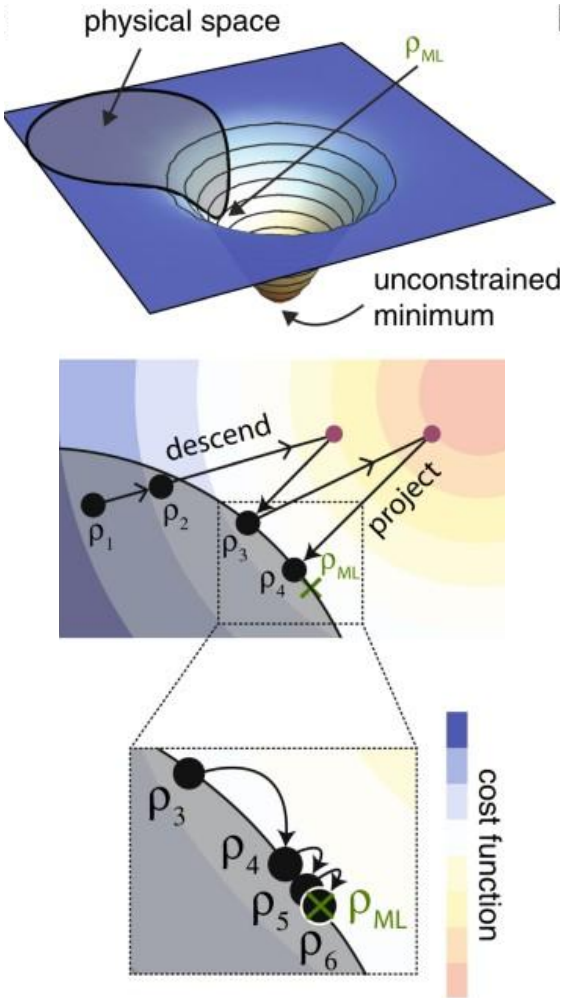


Figure 4.16: Illustration of the parameter space and the physical constraints imposed on it. Furthermore, the optimization process, which alternated between gradient descent and projection into the admissible region, is shown. Figure from [Bol+17].

where the soft-constraint problem of Equation 4.8 becomes a hard-constraint one. Recent advances in neural network optimization demonstrated noticeable successes in many fields using conditional gradient (CG), which leads to Pareto-optimal solutions and eventually to improved generalization [Rav+19a]. Indeed, formulation (Equation 4.8) falls under the category of "scalarization" technique whereas (Equation 4.11) is ϵ-constrained technique. It is known that when the problem is non-convex, ϵ-constrained technique yields Pareto-optimal solutions, whereas the scalarization technique does not [BV04; Rav+19a]. We ensure the fulfillment of the additional derived constraints using CG, where gradient steps rely on a

linear minimization oracle over the set of constraints defined by the additional regularization term.

4.5 Experiments

The application we use here is part of ongoing efforts to develop sustainable approaches in iron and steel industry. The goal is to exploit the iron scale produced by the iron and steel industry in order to obtain a rust-proof paint pigment. This raw material will be used in a defined proportion mixed with a natural iron oxide pigment. We are mainly interested in the study of their physicochemical characteristics [Abe18], and particularly, thermal and mass loss analysis as shown in Figure 4.17.

In this section, we introduce some domain-specific notions that are important to understand the experimental setup. We present the main materials used in our experiments: red pigment and calamine oxide and their mixtures. Then, as the theoretical framework used to control real experiments is based on thermal analysis and kinetics models, we will give a short description of both of them.

4.5.1 Application description

Binary mixture and target material. The application goal is to characterize and to synthesize a new paint pigment based on the calamine oxide and red pigment ensuring desired properties at some given temperatures. *Red Pigment* is a natural form of mineral composed mainly of iron oxide; its individual thermal signature is given in Figure 4.17a. This analysis shows a mass loss which is attributed to the evaporation of water formation of iron hydroxides corresponding to the dissolution of goethite FeO (OH) [ABG17]. *Calamine Oxide* is a steel by-product obtained during continuous casting or heating of slabs and billets. This product is not a sterile waste and may have a meaning of raw materials in its own, which can be valued and marketed. The synthesis of new materials is obtained by the contribution of the calamine in this process by ensuring a sufficient quantity of Fe_2O_3 and increasing the density of the synthesized pigment (Figures 4.17b and 4.17c). In parallel, the goal of this application is to get materials with some desirable qualitative properties, including optical properties (the size of the pigment particles may affect the final appearance of the coated surface: paint can be glossy, matte, or satiny, depending on the particles' size which affects the phenomena of diffusion, reflection, and refraction of light), ferromagnetic properties, etc.

The new material synthesis can be viewed from several theoretical models. Each one uses approved knowledge of the field and predicts the expected theoretical

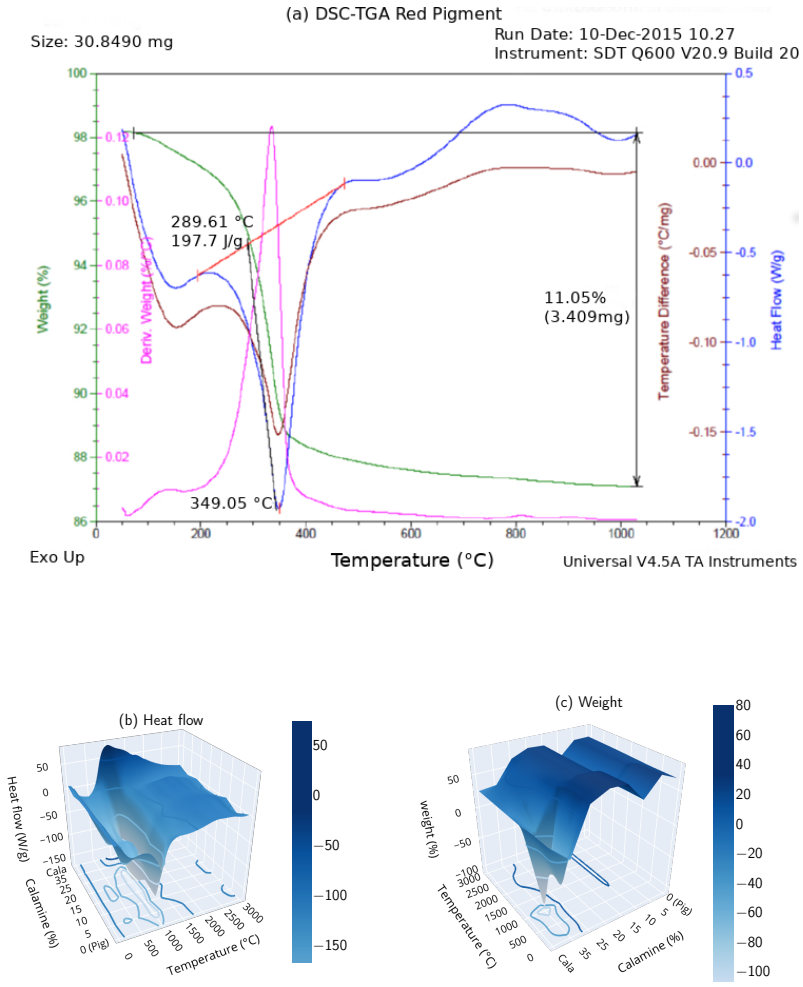


Figure 4.17: Simultaneous thermal and mass loss analysis of (a) red pigment and (b, c) binary mixture of red pigment and additional calamine percentages. The effect of the temperature augmentation on the behavior of the red pigment is shown via weight, derivative weight, temperature difference, and heat flow curves. Further analysis of mass loss, variation of the dissociation reaction enthalpy, and the formation of new phases can be found in Section Thermal Analysis.

trajectories. The most prominent models that we leverage in our proposed method are *thermal* and *kinetics* models.

Dataset description. Our dataset is built from real experiments using the SDT-Q600 version 20.9 thermogravimetric analyzer. Dataset consists of a thermal analysis of raw materials. These were collected with an SDT-Q600 industrial instrument that monitors the calcination of the mixtures continuously. The instrument encompasses a pair of thermocouples within the ceramic beams that provides a direct sample, reference, and differential temperature measurements from ambient to 1500 °C (using a ramp of 40 °C/min). Specifically, various signals are monitored by the instrument, including weight (mg), heat flow (mW), temperature difference (μV), sample purge flow (mL/min), etc. The dynamics of the Nitrogen gas, which constitutes the ambient atmosphere around the mixture, is set to 100 ml/min. The acquisition of the various signals was carried out at a sampling rate of 2 Hz, which is sensitive enough, in these kinds of applications, for capturing temperature and mass trends that may indicate regime changes. In total, 3000 measurement points were obtained for each set of experiments. In addition to the theoretical curves of the red pigment (pig) and the calamine oxide ($cala$) that were obtained separately, we perform calcination of mixtures with various percentages, $p_i \in \{5, 10, 15, 20, 25, 35\}$, of additional calamine oxide.

Training details. We construct neural networks by stacking 3 Fully Connected/ReLU layers with a dropout probability of 0.5 and two regression outputs (for weight and temperature). We optimize the neural architectures, including the number of neurons in each layer, using Bayesian optimization [SLA12] (The complete list of hyperparameters and their range of values can be found in the code repository). The networks are trained for 1000 epochs on the training data and evaluated on the test set. The learning rate is set to 0.0001. We perform train-test splits over different runs by stratifying the learning examples. The model is trained to reconstruct the weight and the temperature state variables simultaneously by minimizing the mean squared loss between the original target signal and the reconstruction provided by the network. In the case of SGD, weights of the neural network are optimized using the Adam algorithm [KB14]. As a reference, we also train a model (referred to as baseline) using the same evaluation setup and a comparable number of parameters but without any additional derived constraint.

4.5.2 Evaluation of the reconstruction process

In our first set of experiments, (1) we evaluate the reconstructions obtained using different configurations of the real experiments based on the setting described in the proposed evaluation protocol; (2) we assess the extent of reconstructions as a function of the distance to the set of validation experiments and the impact of using CG on the fulfillment of the additional constraints; (3) we evaluate the reconstruction performances at specific percentages of additional calamine.

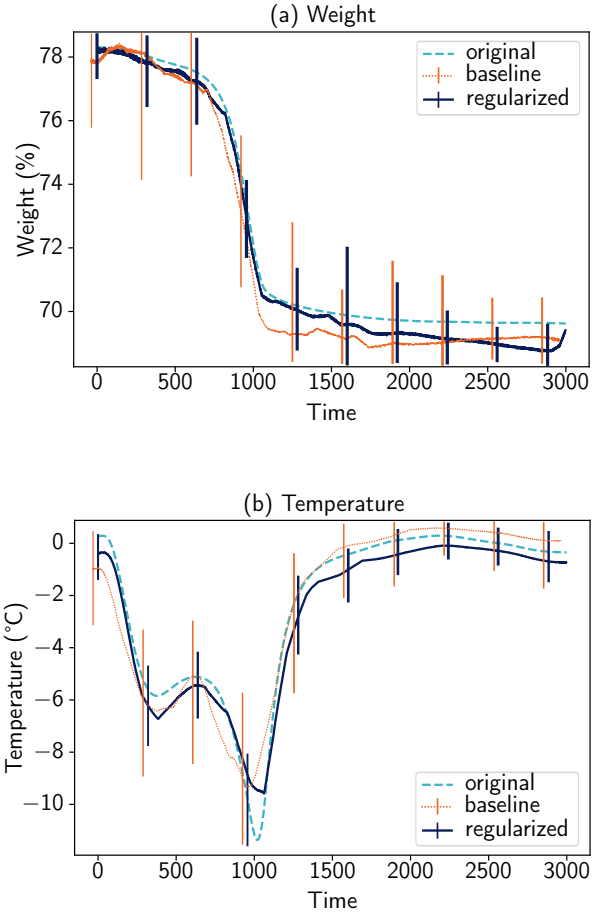


Figure 4.18: Obtained state space reconstructions for (a) weight and (b) temperature. We report reconstructions averaged over all evaluation setups and their corresponding perplexity. As references, we also report the reconstructions obtained (under the same evaluation setups) using the baseline.

Figures 4.18a and 4.18b show the obtained state space reconstructions of weight and temperature, respectively. Obtained reconstructions are averaged over all evaluation settings. We additionally report their corresponding perplexity. These two figures highlight, in particular, the perplexity of the naive approach (baseline). Our approach contributes to a substantial reduction of this perplexity (e.g., 2.76 ± 0.09 vs. 3.29 ± 0.15 for weight; 55.7 vs. 59.4 for temperature). The perplexity here can be related to 2 factors: the spacing of the real experiments; and the presence of phase transitions, especially in the range [250; 1250] for temperature. To verify

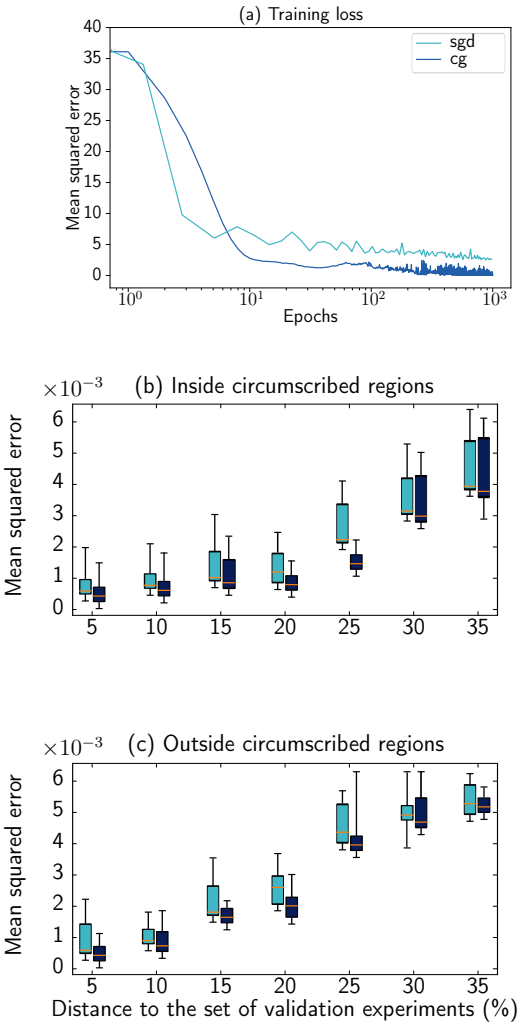


Figure 4.19: Comparing the performances of SGD vs. CG: (a) evolution of the training loss as a function of the number of training epochs; (b and c) the extent of the reconstructions as a function of the distance from the set of training to the set of validation experiments (inside and outside circumscribed regions of the state space, respectively). We repeat the evaluation 10 times with different random seeds and report the median and the best validation performance of the models.

the impact of experiment spacing, we measure the extent of reconstructions as a function of the distance from the set of training to the set of validation experiments. We provide numerical evidence in Figure 4.19 with the evaluation protocol

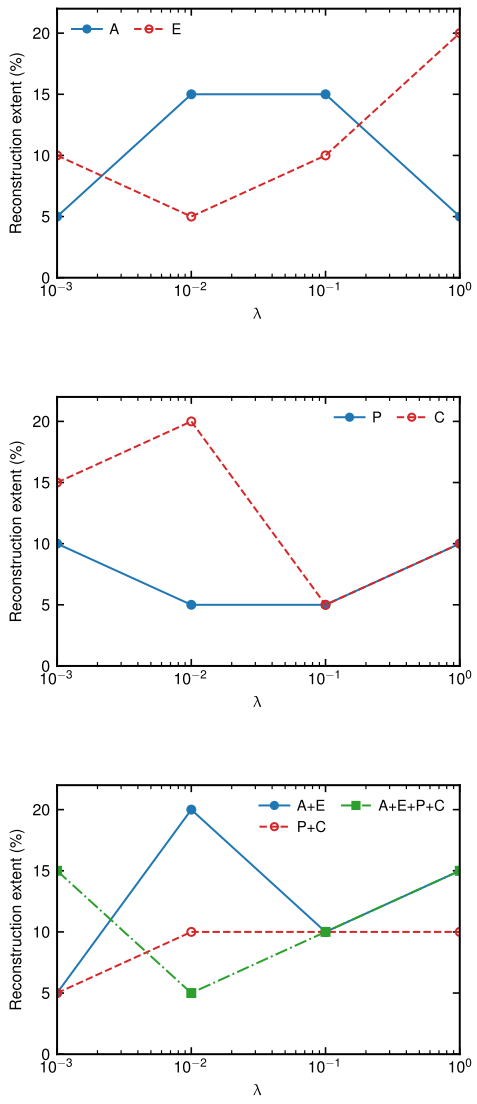


Figure 4.20: Reconstruction extents depending on the control parameter being considered, which is the type of analytical model being used to augment the real experiments.

defined above. We repeated the evaluation setup for 10 times. We can see that until 20%, both inside and outside circumscribed regions, our approach provides controlled perplexity.

Furthermore, Figure 4.19 illustrates also the impact of using CG on the fulfillment of the constraints that are imposed on the models. We can see a noticeable

effect of CG on the reconstruction performances up to an extent of 25% (Figure 4.19b) and 30% (Figure 4.19c). This translates the ability of CG to converge towards solutions that take into account the regularization-like terms, whereas SGD tends to push towards solely satisfying the first term of the cost function at the expense of providing constrained reconstructions. After that extent, we can notice that the performances of CG and SGD are comparable for models trying to extrapolate far away from the real experiments. This could be explained by the fact that the penalty bounds are becoming loose from that point, which does not help the model to reconstruct correctly. Despite the existence of many phase transitions that span all over the state space, our approach is particularly able to reconstruct the weight and temperature states. Even when we reduce the number of real experimental points, the obtained reconstruction quality remains high.

We further investigate the extent of reconstructions at specific percentages of additional calamine oxide. We report the average reconstruction performances over all configurations of the sets of training experiments. It is worth noticing that for some percentages, e.g., reconstructions of temperature curves at 15%, no matter how far apart the set of training experiments are, the reconstructions are satisfactory with or without the addition of analytical models. However, the analytical models contribute substantially to reducing the accompanied perplexity ($0.00192 \pm .00081$ vs. $0.0076 \pm .0023$). On the other hand, for 35%, for example, the reconstruction errors are greater using the baseline model. This could also be explained by the numerous phase transitions that exist around this percentage. In this case, our approach is able to significantly improve upon the baseline model and overall in all percentages both in terms of approximation and perplexity (e.g., $0.00087 \pm .00122$ vs. $0.00477 \pm .0021$ at 15%; $0.00246 \pm .002$ vs. $0.00932 \pm .0056$ at 35%).

4.5.3 Trade-off between real experiments and richness of the domain models

In the previous experiments, we showed the ability of our approach to reconstruct precisely both the portions of the state space delimited by and those falling outside sets of experiments. Here, we evaluate the trade-off between the richness of the domain analytical models, which are plugged into the optimization objective of the learning models via regularization, and the granularity of real experiments.

For this, we compare reconstruction models trained using different configurations of the kinetic and thermal-based regularization-like terms. Precisely, we use the *Arrhenius* and *Eyring* models, as well as the theoretical curves, *pig* and *cala*, to derive these terms. We distinguish a first configuration where the analytical models are each plugged individually to guide the learning models and a second

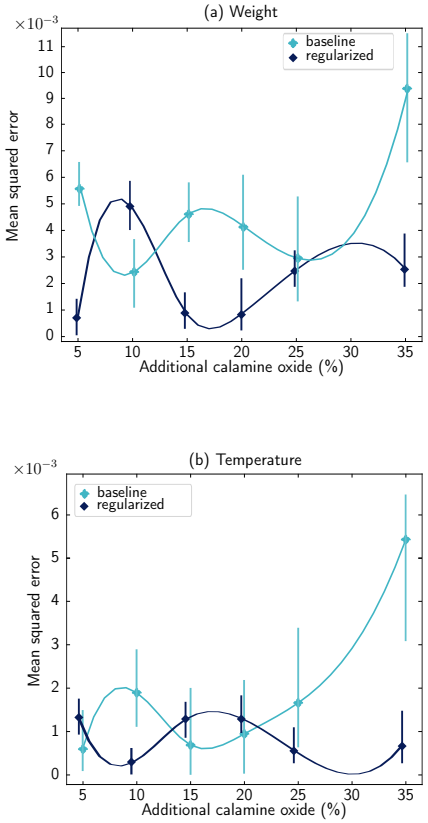


Figure 4.21: Reconstruction performances at specific percentages of additional calamine oxide. We compare the reconstructions, of (a) weight and (b) temperature, obtained using the baseline vs. the regularized models. Results averaged over all possible distances to the set of training experiments.

configuration where we combine them together, i.e., *Arrhenius* and *Eyring* models (A+E), *pig* and *cala* (P+C), and all these models combined together (A+E+P+C). Additionally, we provide the best extent of reconstruction (in %) that was achieved in each configuration. Figure 4.20 summarizes the obtained results.

In both configurations, the analytical models significantly improve the performances of the reconstructions generated by the learning models. Very interestingly, using the theoretical curve of the red pigment, the constructed models are able to get a substantial gain in terms of reconstruction losses. In particular, for $\lambda = 0.01$, we obtain an improvement factor of 10 over the remaining values of λ of the same configuration. On the other hand, the models guided by the theoretical curve of

the calamine outperform those guided by the red pigment (except for $\lambda = 0.01$, but there the difference is smaller than for the other values of λ). This observation shows that some analytical models are more adapted than others, which is further confirmed when we compare the influence of the *Arrhenius* and *Eyring* models on the generated reconstructions. Besides, we can observe that models guided by a combination of A+E outperform both A+E+P+C and, by far, P+C while attaining a reconstruction extent of over 20%.

Overall, these combinations have better reconstruction performances than the baseline or analytical models taken individually since their impact is adapted to different regions of the state space. These results give additional insights into the study of trade-offs between the richness and complexity of domain analytical models and the amount of real experimental data (or sensor measurements in the case of real sensors) needed to train learning models.

4.6 Conclusion

In this chapter, we have looked at different forms of prior knowledge from the domain as well as how to integrate them into the learning process in an explicit way. We focused on the structural constraints on the input data and how they can guide the learning process to reach regions of the parameter space containing satisfactory solutions to learning problems. We proposed two approaches: one that selects learning examples to provide to the learner guided by domain models (§ 4.2) and another that perform augmentation in appropriate regions of the example space, based again on domain knowledge (§ 4.4).

The former approach leverages additional information about the way data sources (or configurations in a broader sense) in a sensing environment are structurally related to each other. Practically, we use invariants to encode the decision boundaries shared across configurations. The latter approach proposes an augmentation guided explicitly by the knowledge of the regions of the example space that require an augmentation. It was formalized in the form of a two-level optimization problem similar and exploits the capacity of this process to exhibit invariant aspects in the learned models and reinforce them.

More generally, the main idea behind these approaches was to ensure that the learner retains important elements throughout the learning process and throughout the tasks they encounter (whether in a continuous or federated framework). The type of elements that we have seen in the above approaches is the portions that remain invariant to the transformations underlying the data distributions. We were able to achieve this through the selection and generation of relevant examples that make such invariant portions emerge and reinforce throughout the learning process. Experiments show promising results in terms of the number of required examples

to learn in heterogeneous and dynamic environments. The proposed approaches were shown to be effective in situations where the generation and transmission of learning examples are constrained, e.g., in federated learning and continual learning settings.

A high-level motivation for this type of approach finds its roots in the concept of *teachability* [6], where people investigate how the order in which material is presented (by a teacher) can lead to qualitatively and quantitatively different learning outcomes (for a student). Indeed, the approaches proposed in this chapter are analogous, to some extent, to the traditional way the teacher and student interact at school in the context of education [Fan+18]. The teacher provides the student with appropriate material about the concept (or set of concepts to learn) depending on the student's level of mastery. In that context, material corresponds to textbooks or exercises and is designed in a way that it targets specific notions that are lacking or need to be reinforced in the student's mind (portions of the decision boundaries in our proposed approaches). In other words, instead of presenting any learning examples and in any order, the learner should deal with examples that correspond to its level of mastery. The presentation of the learning examples can indeed vary according to (i) the sequencing, (ii) the assigned weighting (easiest examples to classify to the hardest), and (iii) the choice of the examples themselves. In curriculum learning approaches, the order is often determined by how easy the learning examples are, but, as one can see from an educational point of view, this involves much more complex aspects.

As mentioned at the beginning of this chapter, various works have analyzed these strategies and tried to explain their performances: in particular, some of them consider these strategies to be a particular form of continuation optimization methods, where a given optimization problem, often complex, is dealt with gradually by starting with simpler versions of the problem until the original problem is solved [AG12; Ben+09]. This is one of the control parameters that can be used to act on the learning process. Indeed, among the existing control parameters, one can find: modifying the hypothesis space (e.g., changing the architecture of a neural network); adjusting the hypothesis space exploration process (e.g., weight decay); modifying the optimization landscape (e.g., acting on the order in which the learning examples are presented). One of the principles underlying the family of approaches we proposed in this chapter is the impact that the way of presenting the learning examples to the learners has on the optimization landscape. If the presentation of the learning examples is done in a suitable manner, this may have the effect of modifying the optimization landscape to make it, in some way, easier to navigate (see Figure 4.22). Identifying the opportune way of showing

[6] A term found in [Rit+07, Chapter 3] and [Fai16] both of which study aspects related to education

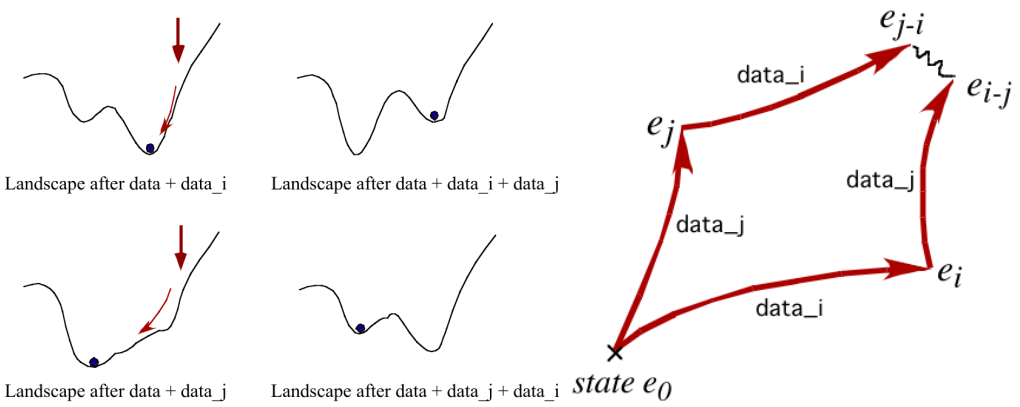


Figure 4.22: The order in which the learning examples are presented to the learner modifies the optimization landscape. Figure from [Cor07].

the learning examples is a crucial challenge in reducing the size of the learning problem. From the perspective of the structural risk minimization framework of Vapnik [Vap92], the ordering of the learning examples is, in a way, introducing a structure to the set of admissible hypotheses.

Finding a good solution (or hypothesis) to the learning problem requires the traversal of the optimization landscape, which could be very difficult. Nevertheless, the very often non-convex optimization landscape is explored using local search heuristics, as simple as gradient descent, achieving remarkable state-of-the-art results. The difficulty of this traversal depends on the properties of the optimization landscape [Dau+14; Li+18; Ahm+19]. Indeed, the optimization landscape might be chaotic with shallower regions of convexity, where the gradients provided by the local search heuristics are likely uninformative [Li+18]. Furthermore, authors in [Dau+14] investigated the prevalence of saddle points in high-dimensional non-convex optimization problems, which may hinder learning and make the optimization procedure take a long time to escape. The curvature of the optimization landscape can also vary rapidly, which makes choosing a step size for the optimization procedure very difficult [Ahm+19].

As we saw, the order in which existing curriculum approaches provide learning examples to the learner depends on their easiness, where very often, the easiest examples go first. This changes the optimization landscape in a particular manner. In our proposed approaches, the selection (as well as the augmentation process) is guided by domain models. Therefore, the modification of the optimization landscape is tightly linked to domain knowledge. Understanding how the optimization landscape behaves and exhibiting its geometric properties may help develop better heuristics that can explore it efficiently. Long lines of research have been dedi-

cated to the challenging problem of establishing such properties [Dau+14; Ge+15; Kaw16; GM17; Li+18]. For example, the optimization landscape of many objective functions has been conjectured to have the geometric property that "all local optima are (approximately) global optima". It is this property that makes local search algorithms perform well on these problems [GM17].

More related to domain knowledge, invariants of the domain or symmetric distributions have an impact on the optimization landscape and shape the way it can be explored. For example, symmetries lead to non-convexity, especially saddle points. In that matter, authors in [ZQW20], for example, highlight that many real-world learning problems exhibiting symmetries have another kind of geometric property: "local minimizers are symmetric copies of a single "ground truth" solution, while other critical points occur at balanced superpositions of symmetric copies of the ground truth and exhibit negative curvature in directions that break the symmetry." The idea is that "symmetries of the observation models become symmetries of the optimization problem." [ZQW20] And ultimately, efficient methods to traverse the optimization landscape can be obtained.

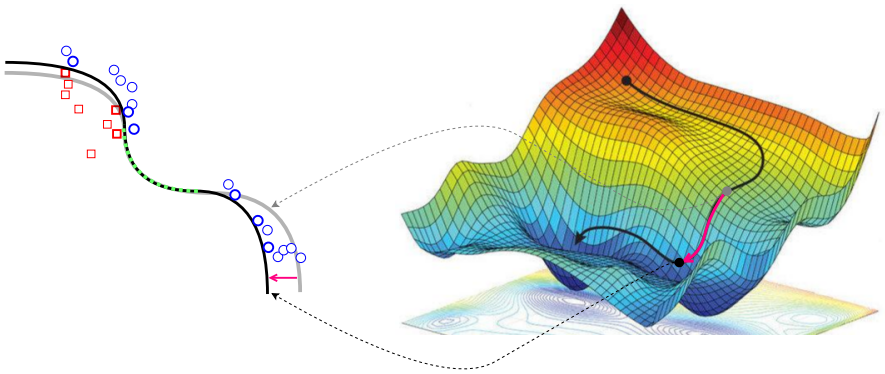


Figure 4.23: The order of presentation of the examples has an essential impact on the optimization landscape. Fine-grained control of this procedure can facilitate the exploration of this landscape and ultimately achieve satisfactory solutions to learning problems. The question that arises is how to ensure that this procedure makes it possible to make inductive leaps of quality in the optimization landscape and not locally circumscribed modifications. In this chapter, we have proposed approaches involving domain knowledge and the invariant portions of decision boundaries to achieve this.

Regarding the traversal of the optimization landscape, alternative approaches to local optimization methods could probably be to take a high-level (or global) standpoint on the optimization problem. Having a high-level standpoint on the

optimization process could be essential in order to make quality inductive leaps and consequently improve learning performance (see Figure 4.23). Indeed, neural networks, for example, encode functions, and a high-level standpoint on the optimization problems could be to reason in terms of the function space rather than the parameter space. Naturally (and very often), in order to optimize a neural network, one makes adjustments to its parameters. Various analyses showed, however, that reasoning in terms of functions is beneficial. Many studies focus on aspects around inductive jumps in the space of functions, in particular, the works of [BRK18; Ber+20] on the distance between models in the space of functions but also, to a certain extent, work around the search for neural architectures (where it is rather the particular architectures that encode functions) [GH19; FC18].

A challenge in this regard is that an adjustment in the parameters does not necessarily translate into a change in the function space. In other words, the relationship between changes in the parameter space and in the function space is not straightforward. For example, as shown by authors in [BRK18], networks cross the function space differently compared to how it is done in the parameter space. Thus a distance of parameters cannot be taken to represent a proportional distance between functions. More appropriate distances are therefore needed. For example, many works, such as [BRK18], propose approaches to calculate distances between functions in well-determined spaces (L^2 Hilbert space or change in a network's output distribution as measured by the Kullback-Leibler divergence) instead of the usual space of weights (ℓ^2). Precisely, authors in [BRK18] argue that the L^2 Hilbert space is useful for analysis and proposed a regularized loss in the context of multi-task learning: $\ell(\theta) = \ell_{\tau_j}(\theta) + \frac{\lambda}{2}\|f_{\theta_{\tau_i}} - f_{\theta_{\tau_j}}\|$, where regularization term is the L^2 distance between the current function $f_{\theta_{\tau_j}}$ and the function after training on task τ_i, $f_{\theta_{\tau_i}}$. To compute the L^2 distance between the current version of the model and the previous one, the authors store a small set of previous examples in working memory (similar to the replay buffer used in continual learning approaches), as well as the model's outputs on those examples. Our proposed approach based on the selection of appropriate learning examples to update the learning model can be considered as a form of optimization in the function space (as opposed to the weight space). The selected examples are assumed (or intended) to update the learning model in meaningful ways, i.e., in accordance with domain knowledge.

Other related works propose to use output regularizers [Xie+16; Per+17], which, as their name suggests, put constraints on the output layer of the models and, as such, penalize various behaviors such as over-confident predictions (often a symptom of overfitting according to [Sze+16]). We investigated in this chapter the integration of domain models via constrained gradient descent, which was used to ensure that the learned models, indeed, satisfy the imposed constraints originating from the domain. The question that remains, however, is the difference between

(i) constraining an optimization space defined by the conjunction of data, data transformations, and loss function (roughly, a posteriori or during the learning process), with (ii) constructing an optimization space that is itself constrained by domain knowledge but a priori. This may seem trivial at first sight, but the two strategies are not totally the same and deserve further investigation (interesting properties may emerge from this distinction). The approaches that we have seen until now could be categorized into the former type of strategies: even if the order in which the learning examples are presented to the learner affects the optimization landscape, this is done in the course of the learning process. An alternative approach that could be categorized in the latter set of strategies is presented in the next chapter and proceeds by structuring the concepts to learn into hierarchies as a prior step before the learning can take place at each level of the hierarchies.

We saw in this chapter how to leverage domain knowledge to guide the selection and augmentation of learning examples. The underlying structures considered in this chapter are assumed to be available. In the next chapter, we will describe different approaches to come up with such structures from data.

Chapter 5

Structuring the learning process guided by the concepts to learn

In this chapter, we propose two approaches that leverage the semantics of the label space for organizing the learning process. The idea is to decompose the learning process into several sub-problems that are easier to solve while maximizing the notion of reuse and sharing (transfer) between these sub-problems. The approaches presented in this chapter are based on [OHA21a] and [OHA22].

In the previous chapter, we were able to see different forms of structures from the domain as well as how to incorporate them into the learning process through the selection and generation of learning examples. We have discussed the pros and cons of the different techniques while looking at their practicality. In particular, we discussed the links with landscape optimization and the impact of data on the shape of the latter. In this chapter, we will take a different perspective than that which is based on the structuring of input data to guide learning. We will rather exploit the structuring of the concepts (or labels) to be learned, in other words, the outputs, and more particularly, the structuring of the concepts into hierarchies. These offer a way to more finely control the sequencing of the learning process in terms of difficulty, starting by learning the characteristics (patterns, motifs, or common points) of groups of concepts (a task that is easier) and then moving towards learning the characteristics of the concepts taken individually. This task being more specific and, therefore, more difficult. In other words, the separation of the groups of concepts is based on coarser characteristics than those useful for separating atomic concepts.

From an application point of view, this perspective is motivated by the ubiquity of dependencies between concepts, particularly in distributed and decentralized applications. The discrepancies between the data sources describing the concepts are

exacerbated in such applications. For example, in IoT applications and particularly in human activity recognition (HAR), some concepts (or activities) naturally intermingle, and the boundaries between these concepts are not clear, e.g., the transition from the concept *walking* to *running* remains blurred and necessitates finer attention. These phenomena are further accentuated by the capabilities of the sensors (sensing models) and the perspectives (views) through which the concepts are captured (e.g., sensor's position in space, position on the body, type of modalities, sensor's characteristics) [AC09; HO20]. Incomplete or redundant perspectives can lead to further confuse the concepts between them and reduce the performance of the learning process. Beyond the dependencies (overlap) relating to the perspectives provided by the deployments of sensors, the learning problems themselves and the concepts which compose them often exhibit intrinsic dependencies [SF11; EOR15]. Furthermore, this point of view is also motivated by the natural link with learning in a student (student) where the concepts should be presented by an increasing degree of difficulty: from the simplest tasks to the most complex. Indeed, We find that some concepts are easier to distinguish when grouped with other concepts than when each one is learned on its own. For instance, if we consider analyzing human activities through the accelerometer or heart rate, it is easier for a given learner to first separate all activities (concepts) into two main classes, e.g., activities involving large movements of the hand versus other activities, instead of separating the finer activities belonging to (or lying within) these two general classes. This general observation shows that inductive biases needed to separate homogeneous groups of concepts recursively give better results and build hierarchical concept structure between concepts.

We explore in this chapter different strategies for structuring the considered concepts into hierarchies such that those very similar concepts are grouped together and tackled by specialized classifiers. The idea is that classifications at different levels of the hierarchy may rely on different features or different combinations of the same features [ZXW11; Yao+19]. Indeed, many real-world classification problems are naturally cast as hierarchical classification problems [CH04; WCB18; Yao+19; ZXW11]. Work on the semantic relationships between categories in a hierarchical structure shows that they are usually of the type generalization/specialization [ZXW11]. In other words, the lower-level categories are supposed to have the same general properties as the higher-level categories plus additional, more specific properties. This can be naturally cast as a meta-learning problem, where one can pursue a bi-level processing with meta-parameters shared across the hierarchies and various levels of group-specific parameters. We will start in Section 5.1 by formalizing this problem and providing a literature background around approaches that are intended to characterize and compute how tasks are related to each other.

Unlike the previous chapter, where the structures underlying the input data are available a priori, here we build these structures (or, more precisely, hierarchies) automatically from the data. This is the bias learning part, or concept group bias to be more precise: a form of feature learning that is refined on several levels (from the most general to the most specific) see Figure **??**. For this, we have proposed two approaches: one called top-down, which is based on clustering and the decomposition of groups of concepts (§ **??**) and the other bottom-up, which is based on the affinity of transfer and the composition of concepts starting from those which exhibit a weak affinity to the transfer until the strongest (§ **??**).

In Section 5.2, We propose two novel measures (dispersion and cohesion) to assess the quality of clustering solutions regarding concept separability. We propose an efficient clustering-based classification approach combined with training strategies that leverage the tree structure to improve the learning process. The components of the proposed approach, including the theoretical complexity of the hierarchical learning problem, which is substantially reduced, are analyzed. Extensive experiments are conducted on three HAR datasets to assess the effectiveness and efficiency of our proposed approach (§ 5.3). The notion of inductive biases inheritance in the hierarchy of concepts being derived is also investigated. Furthermore, we empirically analyze the notion of intrinsic concept dependency and its relativity w.r.t. the various perspectives (views) provided by the sensors deployments and how the proposed measures capture these two kinds of dependency.

In Section 5.4, we propose an approach based on transfer affinity to determine an optimal organization of the concepts that improve both learning performances and accelerates the learning process; We leverage for this a powerful technique based on transfer which showed interesting empirical properties in various domains [Zam+18; PRS19]. Taking a bottom-up approach allows us to leverage learning the complete hierarchy (including the classifiers assigned to each non-leaf node) incrementally by reusing what was learned on the way. Extensive experiments show the effectiveness of organizing the learning process. We noticeably get a substantial improvement in recognition performances over a baseline that uses a flat classification setting; we perform a comprehensive comparative analysis of the various stages of our approach, which raises interesting questions about concept dependencies and the required amount of supervision (§ 5.5). The approaches presented in this chapter are based on the following works [OHA21a; OHA22].

5.1 Problem Statement

The main idea explored in this chapter comes from the fact that the concepts to be learned are not totally independent, as is the case in human activity recognition where, for example, learning the concept *running* is closer to learning the concept

walking than learning the concept *still*. Thus, grouping some concepts to learn them against other groups of concepts, using more adapted biases or characteristics, can considerably improve the learning process quality for each concept.

Let's consider $Y = \{Y_1, \ldots, Y_n\}$ a set of atomic concepts (or classes) to learn. In this chapter, we show that for a given specific *a priori* knowledge on these concepts, the quality of the learned hypothesis improves by grouping the concepts recursively. We assume that atomic concepts are not decomposable, i.e., $\forall i \neq j \in \{1, \ldots, n\}, Y_i \not\subset Y_j$), and any group of concepts GY_i is a subset of Y. Since the atomic concepts have partial dependencies in many cases, a top-down approach tries to structure the atomic concepts into different combinations and based on different biases. It gives a better loss function than the one used in the flat case. This idea is close to the decision tree [Qui86] but more general. It is applied to the separability of the groups of concepts rather than to atomic concepts. This formalization follows the idea presented in [Kos+15], which defines a three-dimensional setting: (1) single-label classification as opposed to multi-label classification; (2) concepts are organized into trees as opposed to directed acyclic graphs; (3) instances are classified into leaves (mandatory leaf node prediction [SF11]), as opposed to the setting where instances can be classified into any node of the hierarchy.

Therefore, the problem at hand is twice difficult as we have to first find the most appropriate hierarchical structure and, second, find optimal learners assigned to the nodes of the hierarchical structure. Some works have tackled this problem by exploiting a priori knowledge and structures of the domain [SBH20; Sch+20]. However, such a priori knowledge is not always available. One of the main problems to solve, in this case, is finding the best tree structure of groups of concepts to learn together in order to optimize the learning rate of each atomic concept. A naive approach consists in building all the combinations of concepts to check for which groups of classes the quality of the learning is optimal and to start again recursively this approach until the concepts are totally separated from each other. However, this approach faces a combinatorial explosion of the number of cases that should be treated.

To better illustrate the complexity of this problem, we propose a recurrence relation involving binomial coefficients for calculating the total number of tree hierarchies for a total number of n concepts.

Theorem 1. Let $L(n)$ be the total number of trees for the n atomic concepts. The search space size for these concepts satisfies a recurrence relation defined as:

$$L(n) = \binom{n-1}{n-2} L(n-1)L(1) + 2\sum_{i=0}^{n-3} \binom{n}{i} L(i+1)L(n-i-1)$$

Because of the exponential size of the search space, the exact approaches cannot tackle this problem in terms of time/space complexity for large sets of (fine or coarse-grained) concepts like those featured by the SHL dataset [Gjo+18] (described in Section 4.3), which we consider throughout this chapter as a running example to illustrate the problem formulation and the proposed approach on a concrete real-world example. In this dataset for example, with 8 coarse-grained concepts, the size of the search space is $L(8) = 660,032$.

To take advantage of the power of this search space traversal approach and to avoid combinatorial explosion, we propose data-driven approaches for selecting the best concept structuring.

Concept structuring as a meta-learning problem. The case of concept hierarchy is an instance of the meta-learning problem and as such it can be seen in several ways: (i) as illustrated in Figure 5.1, where the upper level corresponds to the stages of construction of the most suitable (optimal) hierarchy, the lower level consisting of a process whose final goal is to adapt the weights of (or what is learned by) each of the nodes of the hierarchy; (ii) the upper level does not correspond to the construction of an optimal hierarchy but is subdivided into several other levels which correspond to the different levels (groupings of concepts) of a given hierarchy (available a priori or built beforehand). In each of these levels, bias learning takes place on a group of concepts which puts the following level in a good position to learn the bias of the level which follows it up to the level of atomic concepts. In both cases, the higher level(s) correspond(s) to bias learning.

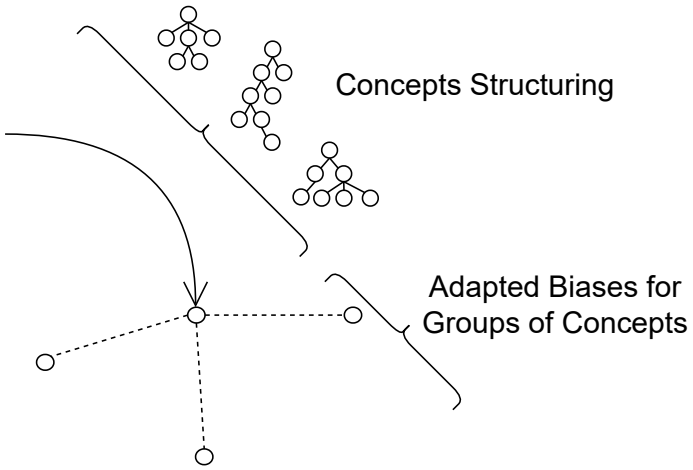


Figure 5.1: Optimizing for more adapted concepts structuring to tackle group biases.

Computing task-relatedness: a literature overview. As we already discussed in the literature review (Chapter 2), various works in the literature propose approaches intended to characterize how tasks are structurally related to each other. The fundamental notion used to make such an analysis and characterize how tasks are related structurally is the measure of task similarity (or task-relatedness). The characterization of such structures has different purposes, e.g., in the context of multi-task learning, the structures are used to find out how tasks transfer to each other [Zam+18], which tasks should and should not be learned together in one network when employing multi-task learning [Sta+20], and devise cluster-specific meta-initializations in the case of GBML approaches [Jer+19; Zho+21b]. Overall, the idea is that after exhibiting the notion of task-relatedness, one can perform, for example, task-clustering in order to devise, in the case of GBML approaches, task-cluster-specific initialization rather than a unique initialization for all tasks, which could be inefficient when tasks are slightly distant to each other [Yao+19].

Existing approaches often measure task similarity by computing similarity scores between tasks either via modeling their data-generating process or leveraging semantic information in the label space. In approaches based on modeling the data-generating process, the computation of the task similarity relies heavily on task-specific models (or networks) such as auto-encoders trained exclusively on a given task used then to quantify its relatedness with other tasks. For example, Taskonomy [Zam+18] proposes a computational approach for modeling the structure of the space of computer-vision tasks, such as texture recognition, semantic segmentation, re-shading, colorization, etc. A task-specific auto-encoder is computed for each of these tasks, and the transfer-affinity of these auto-encoders to other tasks is used to quantify task relatedness. Task2Vec [Ach+19] projects (or embed) the parameters of a task-specific model into a (lower-dimensional) latent space, abstracted from information regarding the number of classes and class label semantics contained in that given task. While Task2Vec uses a pre-trained network to embed a given task into the latent space, the probabilistic task modeling approach proposed in [NDC21] represents a task by a variational distribution of Gaussian task-theme mixture without the need for a pre-trained network (See Figure 5.2 where a given task is represented in the latent task-theme simplex according to the inferred task-theme mixture vectors). In a slightly different fashion, task similarity is computed by looking at the semantics of the target classes in order to capture class dependencies. For example, authors in [Jia+18] try to capture the class dependencies and make use of structured information provided by the label space, such as the images of *cat* and *dog* being closer than *cat* and *truck*. Similarly, Tran, Nguyen, and Hassner in [TNH19] measure task similarity by examining the correlation of the label distributions between the tasks of interest. In a reinforcement learning setting, authors in [Kum+21] generate learning environments using

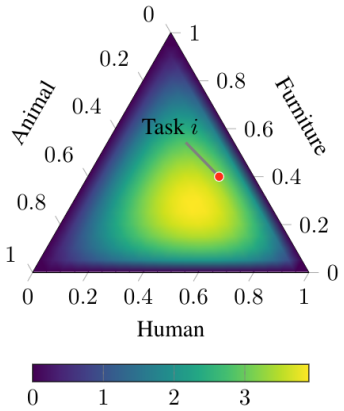


Figure 5.2: From [NDC21]: tasks are represented in a task-theme simplex by a 3-dimensional mixture vector.

explicit rule-based structure generators, i.e., simple rules comparable to the task-theme of Nguyen, Do, and Carneiro, control the recursive composition of learning environments. This way, the authors know exactly the task environment generating process and can evaluate exactly how meta-learning behaves precisely on a broad family of structured tasks.

5.2 Clustering-Based Concepts Structuring

In order to implement our hierarchical-based model for predicting different concepts, we follow two directions: (i) organize the considered concepts into hierarchies such that the learning process accounts for the dependencies existing among these concepts; (ii) characterize optimal classifiers that are associated to each non-leaf node of the hierarchies. Structuring the concepts can be performed using two different approaches: a *top-down* approach, where we seek to decompose the learning process; and a *bottom-up* approach, where the specialized models are grouped together based on their affinities. In this part, we start by describing an approach that follows the former direction. We propose an efficient clustering-based classification approach combined with training (or sample selection) strategies that leverage the tree structure to improve the learning process. To overcome the complexity limitation stemming from the optimal hierarchy construction, we propose an original approach combining clustering and classification of groups of concepts based on two original measures. Precisely, we propose two novel measures (dispersion and cohesion) to assess the quality of clustering solutions regarding concept separability. These measures are optimized throughout the process until the derivation of an optimal learning hierarchy. Furthermore, we design a set of training strate-

gies inspired by curriculum learning that leverage the structural organization of the concepts. The proposed training strategies are specifically designed to leverage the hierarchical structure of the learning process and reduce the amount of supervision required in low-data regimes. It allows a substantial decrease in the number of required learning examples in order to achieve comparable, sometimes better recognition performances compared to the full-data regime and flat classification setting. In this section, we detail the different parts of our approach, which are illustrated in Figure 5.3.

5.2.1 Dispersion and cohesion measures

Let's consider $\mathcal{C} = \{C_1, \ldots, C_m\}$, a clustering result (or solution) obtained in an unsupervised setting (using only the input features X of the instances). Instances of the same concept may be grouped in distinct clusters of the clustering solution. This clustering result can be represented as $G = (Y, \mathcal{C}, E)$ a bipartite graph whose partition has the parts Y (the classification domain or label space) and $\mathcal{C}$ (the clustering domain), with E denoting the edges of the graph (see Figure 5.3). Each edge $e_{ij} \in E$ represents the percentage of the instances from the input space X in class Y_i, properly covered by the cluster C_j. As a consequence the basic normalization property holds: $\forall\ 1 \leq i \leq n,\ \sum_{j=1}^{m} e_{ij} = 1$.

Clustering on the running example. Let's consider a small subset from the SHL dataset containing 365 instances distributed as follows: *still* (Y_1): 40, *walk* (Y_2): 55, *run* (Y_3): 51, *bike* (Y_4): 43, *car* (Y_5): 22, *bus* (Y_6): 25, *train* (Y_7): 62, *subway* (Y_8): 67. Table 5.1 illustrates the distribution of the instances within a single clustering solution.

Clust#	*still*	*walk*	*run*	*bike*	*car*	*bus*	*train*	*subway*	Cohes.
C_1	15	0	3	12	1	3	0	6	0.20
C_2	8	50	45	3	2	2	6	7	0.42
C_3	12	3	1	19	1	0	5	9	0.228
C_4	5	2	2	9	18	20	51	45	0.376
Disp.	0.75	1	1	0.75	1	1	1	1	

Table 5.1: Distribution of the instances within a clustering solution containing 4 clusters. The corresponding cohesion (cohes.) and dispersion (disp.) scores are depicted.

We define two measures, namely *dispersion* and *cohesion* between clusters and classes. The main idea is to evaluate how a given clustering result (obtained in an

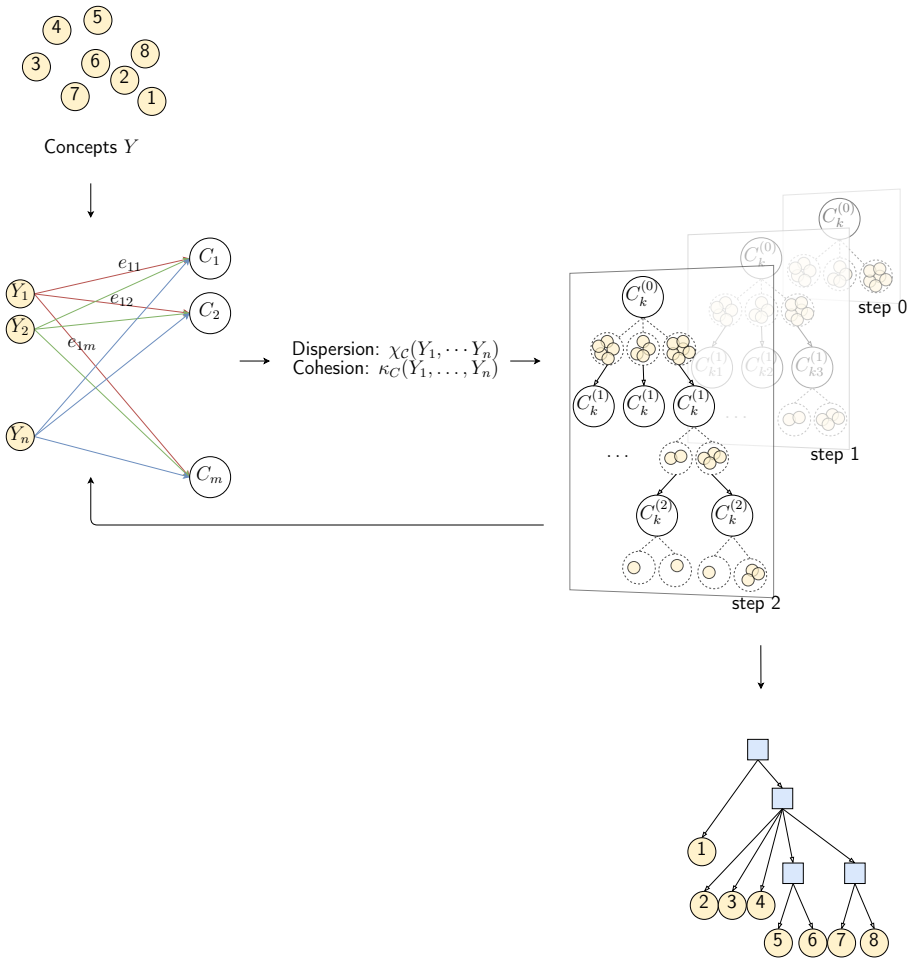


Figure 5.3: The framework of the proposed approach. Based on the dispersion and cohesion score obtained for each cluster, the best clustering solution is selected (step 0), and the process is repeated recursively on each group of concepts within the selected clustering solution (subsequent steps 1, 2, etc.). The process ends as soon as we get individual concepts on the leaves of the decomposition hierarchy. The final hierarchy is being assigned with specialized learners at every non-leaf node. These learners will be trained on the groups of concepts within their descendant leaves.

unsupervised manner) captures the dependencies between the considered concepts (or the assigned labels according to the labeling process used in the dataset). The goal is to use the clustering result to select the best groups of concepts in a way that they can be separated effectively by means of concepts cohesion and dispersion. The subsequent learning steps are applied on these groups of concepts.

Definition 5.2.1 (Dispersion χ)**.** The dispersion of a class Y_i related to a cluster C_j denoted as χ_{ij} defines how the cluster C_j represents the class Y_i. χ_{ij} measures the distribution of the instances labeled by the class Y_i in the $C_j \in \mathcal{C}$ clustering.

There exist different approaches for defining the class distribution in each cluster. However, in our proposed approach, we consider the basic one $\chi_{ij} = e_{ij}$. We denote this distribution with χ_{ij}. Consequently, the Y_i instances distribution w.r.t. the $\mathcal{C}$ clustering should satisfy the following boundaries for the worst and the best cases:

$$\chi_{\mathcal{C}}(Y_i) = \begin{cases} 0 & \text{if } \forall j \in \{1, \ldots, m\} \ \chi_{ij} < \frac{1}{m} + \epsilon \\ 1 & \text{if } \exists j \in \{1, \ldots, m\} \ \chi_{ij} \geq 1 - \epsilon \end{cases}$$

where ϵ is the given small value. If for each class Y_i, $\chi_{\mathcal{C}}(Y_i) = 0$, then the *dispersion* is total and no cluster represents this class. However, if $\chi_{\mathcal{C}}(Y_i) = 1$ because of $\chi_{iw} \sim 1$ then the cluster C_w represents totally the class Y_i.

The *dispersion* measure can be computed in several ways. In fact, if a given concept is represented by a clustering solution, it can be measured using statistical or *a priori* known properties. We propose here a simple measure defined as follows. Assume a class Y_i and a cluster C_j in the clustering result $\mathcal{C}$, for a given threshold α are given. We define R as an auxiliary measure for the *dispersion* as following:

$$R(Y_i, C_j) = \begin{cases} 1 & \chi_{ij} > \frac{\alpha}{m}, (\text{Assume } 1 \leq \alpha \leq m) \\ 0 & \text{otherwise} \end{cases}$$

With these simplifications, we can define the dispersion measure between the clustering result $\mathcal{C}$ and a given class Y_i as:

$$\chi_{\mathcal{C}}(Y_i) = \begin{cases} 1 - \frac{\sum_{j=1}^{j=m} R(Y_i, C_j) - 1}{m} & \text{if } \sum_{j=1}^{j=m} R(Y_i, C_j) \neq 0 \\ 0 & \text{Otherwise} \end{cases} \tag{5.1}$$

And finally, the dispersion between the classification result and the clustering one can be defined as follow:

$$\chi_{\mathcal{C}}(Y_1, \cdots Y_n) = \frac{1}{n} \sum_{i=1}^{n} \chi_{\mathcal{C}}(Y_i) \tag{5.2}$$

Dispersion on the running example. In Table 5.1, the *dispersion* score of each given concept w.r.t. the clustering results is computed using equation 5.1 and $\alpha = 1$. Based on the table, concepts still and bike have a dispersion score less than 1, because their instances are distributed widely across the different clusters than it is the case for the other concepts. The lower the dispersion scores are, the more difficult to handle the next level of the hierarchy. According to equation 5.2, the total dispersion is $\chi_{\mathcal{C}}(Y_1, \ldots, Y_8) = 0.937$.

Definition 5.2.2 (Cohesion κ)**.** The cohesion of classes $Y_1, \ldots, Y_n$ w.r.t. to a given clustering $\mathcal{C}$ measures the co-appearance of classes together in each cluster.

This measure satisfies the following conditions for the worst and best cases:

$$\kappa_{\mathcal{C}}(Y_1, .., Y_n) =$$
$$\begin{cases} 1 & \text{if } \forall c, d, i \in \{1, .., k\} \;\; |\chi_{ic} - \chi_{id}| = \pm\epsilon \\ 0 & \text{if } \forall c, d \in \{1, .., k\}, \exists i \in \{1, .., m\} \; |\chi_{ic} - \chi_{id}| = 1 \pm \epsilon \end{cases}$$

where ϵ is a given small value. From a statistical point of view, there exist several possibilities to compute the *dispersion* measure. The following one gives the empirical and simplest one. For a given cluster $C_l \in \mathcal{C}$ (where $|\mathcal{C}| = m$), the cohesion of two classes is computed as: $\kappa_{C_l}(Y_i, Y_j) = \dfrac{\min(\chi_{il}, \chi_{jl})}{\max(\chi_{il}, \chi_{jl})}$. Accordingly, the simplest cohesion expression between two given classes can be written:

$$\kappa_{\mathcal{C}}(Y_i, Y_j) = \frac{\sum_{l=1}^{l=m} \kappa_{C_l}(Y_i, Y_j)}{m} \tag{5.3}$$

And finally, the *cohesion* of a given set of concepts as $\{Y_1, \ldots, Y_i\}$ w.r.t. a cluster $C \in \mathcal{C}$ and clustering $\mathcal{C}$ (where $|\mathcal{C}| = m$), are computed as following respectively:

$$\kappa_{C_j}(Y_1, \ldots, Y_i) = \frac{1}{i(i-1)} \sum_{k=1}^{i-1} \sum_{l=k+1}^{i} \kappa_{C_j}(Y_k, Y_l)$$
$$\kappa_{C}(Y_1, .., Y_i) = \frac{1}{m} \sum_{l=1}^{l=m} \frac{1}{i(i-1)} \sum_{k=1}^{i-1} \sum_{j=k+1}^{i} \kappa_{C_l}(Y_k, Y_j) \tag{5.4}$$

Cohesion on the running example. The pairwise cohesion scores for the given clustering example $\kappa_{\mathcal{C}}(Y_i, Y_j)$, can be computed using equation 5.3 as in Table 5.1. The cohesion score of all concepts w.r.t. the clusters, i.e. $\kappa_{C_i}(Y_1, \ldots, Y_8)$ is computed in Table 5.1. The table 5.1 also shows that for this application, it is interesting to learn the following concepts together: still and bike (Clusters C_1 and C_3), walk and run (C_2), and car, bus, train, and subway (C_4). This corresponds to clear semantic biases learned during the clustering step and not explicitly introduced.

5.2.2 Hierarchy derivation and optimization

Thanks to the two proposed measures, it is no longer necessary to enumerate and evaluate all the possible groupings of the search space. This task is delegated to the clustering problem. Therefore, the problem can be reformulated as the search for the best clustering that generates the best grouping of classes. Algorithm 5 describes the recursive process of hierarchy construction from the set of concepts and annotated training examples. It proceeds recursively: given the set of annotated examples X and the set of concepts Y considered at a given node of the hierarchy (starting from the root), the algorithm computes different clustering solutions for a varying number of clusters (from 2 to $|Y| - 1$, the two other extremes being obviously useless). To select the best clustering solution, a natural optimization model based on the two proposed measures can be stated as:

$$\max_{\mathcal{C}} \ \gamma_1 * \chi_{\mathcal{C}}(Y_1, \ldots, Y_n) + \gamma_2 * \kappa_{\mathcal{C}}(Y_1, \ldots, Y_n) \tag{5.5}$$

where γ_1 and γ_2 are additional parameters controlling the trade-off between dispersion and cohesion. This optimization model depends on the selected clustering method and its related distance measure.

5.2.3 Leveraging the hierarchy for efficient training

The non-leaf nodes of the derived hierarchy are assigned with learners trained to discriminate between the concepts or groups of concepts found within their descendant leaves. This implies a bi-level optimization problem with $\mathcal{C}$ (the clustering solution at each step) and θ_t (the weights assigned to each non-leaf node t of the derived hierarchy) as the inner optimization problem. Evaluating the learners' weights exactly can be prohibitive due to the expensive inner optimization. Here we propose a simple approximation scheme. We take advantage of the structuring of learners and the inheritance property of inductive biases in hierarchies to effectively drive the learning process by circumscribing the search space for each group of concepts. The idea is to approximate the weights by selecting the most appropriate learning examples to train with the learners of the subsequent levels in the hierarchy, without solving the inner optimization completely util convergence. We investigate for this two strategies that are designed to improve the learning process, namely, (1) *boosting* strategy: the hard examples are weighted so that the learners located in the descendant nodes focus on them; (2) *student-teacher* strategy [HVD15]: the easy-to-classify examples are selected for training the subsequent learners. We use an additional parameter (temperature $T \in (0, 1)$) which decides how hard or easy it is to classify the examples. Algorithm 6 details the learning process in each node of the derived hierarchy.

Algorithm 5: computeHierarchy

Input : (i) $\{(x_i, y_i)\}_{i=1}^{N}$ set of annotated training examples; (ii) $Y = \{Y_1, \ldots, Y_n\}$ denotes the set of concepts; (iii) Distance measure D to compute the linkage

$\mathcal{D} \leftarrow \{\ \}$
for $t \in 2, \ldots, |Y| - 1$ **do**
  $\mathcal{C} = \{C^{(1)}, \ldots, C^{(t)}\} \leftarrow \text{cluster}(X,\ D)$
  Compute dispersion $\chi_{\mathcal{C}}(Y)$; % *using Eqn. 5.2*
  Compute cohesion $\kappa_{\mathcal{C}}(Y)$; % *using Eqn. 5.4*
  $\mathcal{D} \leftarrow \mathcal{D} \cup \{(\mathcal{C}, \chi_{\mathcal{C}} - \kappa_{\mathcal{C}})\}$
end
$\mathcal{C}^* \leftarrow \text{bestClustering}(\mathcal{D})$; % *using Eqn. 5.5*
foreach $C \in \mathcal{C}^*$ **do**
  $\mathcal{A} \leftarrow \text{getClasses}(C)$
  $\mathcal{X} \leftarrow \text{getData}(Y)$
  if $|\mathcal{A}| = 1$ **then**
    $Child_i \leftarrow Y$; % *i^{th} child of the current node*
  else
    $Child_i \leftarrow \text{computeHierarchy}(X, Y)$
  end
end

Result: Hierarchy $\mathcal{T}$

Regarding the class predictions, in classical multi-label classification settings, these can be done in non-leaf nodes [BK12; SF11]. In our case, we use leaf-mandatory classification, i.e., the examples are assigned to an atomic concept (leaf of the hierarchy). Algorithm 7 describes how predictions are performed given the trained hierarchy, $\mathcal{T}_{\theta_1^*, \ldots, \theta_t^*}$, obtained using Algorithm 6.

5.3 Experiments

The empirical evaluation of our approach is organized into three axes: (1) we evaluate the recognition performances of the derived hierarchies (§ 5.5.1); (2) we evaluate the impact of the proposed measures on the derived hierarchies and the separability of the considered concepts (§ 5.3.3); finally, (3) we provide a preliminary assessment of the interplay of inductive biases inside the derived hierarchies via the analysis of the importance of the learners' hyperparameters (§ 5.3.4).

Algorithm 6: Hierarchy training

Input : (i) $\{(x_i, y_i)\}_{i=1}^{N}$ set of annotated training examples

$\mathcal{T} \leftarrow$ computeHierarchy(X, Y)

$\mathcal{T}_{\theta_1,\ldots,\theta_t} \leftarrow$ initialize() ; % *Initialize the weights of the learners assigned to the hierarchy*

$\mathcal{S}_1, \ldots, \mathcal{S}_t \leftarrow \emptyset$; % *Sets of easy/hard-to-classify examples of each learner of the hierarchy*

while *not done* **do**

  foreach θ_t **do**

    Let the super-scripted concepts, $y^{(t)} \in Y^{(t)}$, be those grouped in node t

    if $\mathcal{S}_t \neq \emptyset$ **then**

      Sample mini-batch from $\{(x_i, y_i^{(t)})\}_{i=1}^{N_t}$

    else

      Sample mini-batch from $\mathcal{S}_t$

    end

    Evaluate $\nabla_{\theta_t} \ell(\theta_t)$ with respect to the mini-batch

    Compute adapted parameters with gradient descent: $\theta_t' = \theta_t - \eta \nabla_{\theta_t} \ell(\theta_t)$

    pred $\leftarrow \theta_t'$.predict$\left(\{x_i\}_{i=1}^{N_t}\right)$; % *Make predictions with the newly adapted parameters*

    foreach *prediction* $pred_i \in pred$ **do**

      if $H(pred_i) < T$ **then**

        ; % *T is a temperature parameter. The lower the entropy of the predictions, the higher the confidence and easy-to-classify the example*

        $\mathcal{S}_{t+1} \leftarrow \mathcal{S}_{t+1} \cup (X_i, A_i^{(t)})$

      end

      ; % *Alternatively use $H(pred_i) \geq T$ (the higher the entropy, the lower the confidence), for the hard-to-classify case*

    end

  end

end

Result: Trained hierarchy $\mathcal{T}_{\theta_1^*,\ldots,\theta_t^*}$

5.3.1 Experimental Setup

Representative related datasets. We evaluate the proposed approach mainly on the SHL dataset (see § 4.3 for details). For comparison, we also evaluate our

Algorithm 7: Prediction

Input : (i) x an example to be classified into one of the atomic concepts; (ii) $\mathcal{T}$ the hierarchy of concepts

$i \leftarrow 0$
Let $Child_{\mathcal{T}}(i)$ be the set of children for the node i in the hierarchy $\mathcal{T}$
while $Child_{\mathcal{T}}(i) \neq \emptyset$ **do**
  $i \leftarrow \text{argmax}_{j \in Child_{\mathcal{T}}(i)} \theta_j(x)$
end

Result: Leaf node i corresponding to an atomic concept

proposed approach on two additional representative datasets, the *USC-HAD* and *HTC-TMD*.

- *USC-HAD* [ZS12] containing body-motion modalities of 12 daily activities collected from 14 subjects (7 male, 7 female) using MotionNode, a 6-DOF inertial measurement unit, that integrates a 3-axis accelerometer, 3-axis gyroscope, and a 3-axis magnetometer;
- *HTC-TMD* [Yu+14] containing accelerometer, gyroscope, and magnetometer data all sampled at 30Hz from smartphone built-in sensors in the context of energy footprint reduction;

Baselines. we evaluate the flat classification setting using neural networks, which constitute our baseline for the rest of the empirical evaluations. To compare our baseline with the proposed hierarchical model, we make sure to get the same complexity, i.e., a comparable number of parameters as the largest hierarchies, while including the weights of the learners. We also use Bayesian optimization based on Gaussian processes as surrogate models to select the optimal hyperparameters of the baseline model [SLA12]. In addition, we compare our proposed approach with the closely related baselines from the HAR literature: DeepConvLSTM [OR16], DeepSense [Yao+17], and AttnSense [Ma+19] (a detailed description of these approaches was provided in Section 4.3.1).

Evaluation and neighborhood bias. Model evaluation based on cross-validation usually relies on a random partitioning process. The random partitioning used in the case of segmented time series introduces a neighborhood bias [HP15]. This bias consists of the high probability that adjacent and overlapping sequences (which are typically obtained with a segmentation process and that share a lot of characteristics) fall into training and validation folds at the same time. This leads to an overestimation of the validation results and goes often disregarded in the

Model	USC-HAD	HTC-TMD	SHL
DeepConvLSTM	65.8±.0028	68.2±.0016	65.3±.012
DeepSense	67.0±.017	68.5±.0032	66.5±.005
AttnSense	68.5±.04	70.1±.005	68.4±.002
Feature fusion	67.2±.001	69.2±.0074	66.8±.0042
Corr. align.	69.5±.004	70.5±.0026	69.1±.06
Proposed	71.8±.001	74.5±.0017	73.7±.006

Table 5.2: Recognition performances of various state-of-the-art models on different representative related datasets.

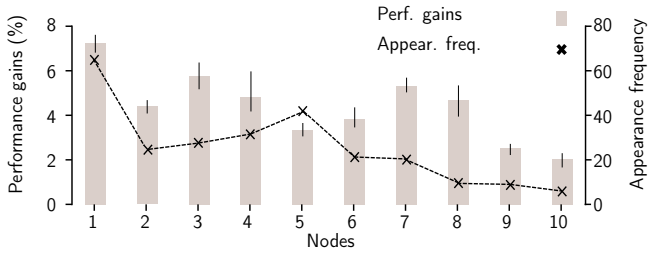


Figure 5.4: Per-node performance gains are averaged over the entire derived architectures (similar nodes are grouped, and their performances are averaged). The appearance frequency of the nodes is also illustrated.

literature. To alleviate the overestimation problem, we rely in our experiments on the meta-segmented partitioning approach proposed in [HP15], which tries to circumvent this bias by first grouping adjacent frames into meta-segments of a given size. These meta-segments are then distributed on each fold.

5.3.2 Performances of the derived hierarchies

Table 5.2 compare the recognition performances obtained with the baseline models on the considered representative datasets. As shown in the table, our proposed approach performs well on the three considered datasets. Note also that performance of the related baselines as reported in the literature confirm the significant issues, analyzed in [HP15], when using regular cross-validation which are likely leading to overly optimistic performance

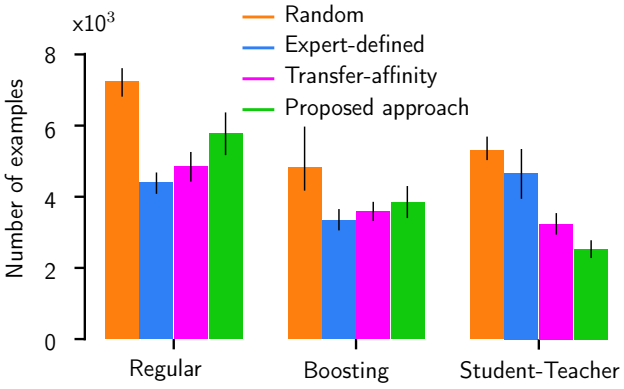


Figure 5.5: The amount of supervision used while training the learners of the hierarchies with the proposed training strategies.

Training the learners assigned to the hierarchy. Figure 5.5 shows the resulting per-node performances averaged over the entire derived hierarchies, i.e., how accurately the learners assigned to the non-leaf nodes can predict the correct groups of concepts associated with them. Each bar in the figure represents the gained accuracy of each node in our hierarchical approach. For example, the 8th bar corresponds to the concepts 2:*walk*-3:*run*-4:*bike* grouped together. Figure 5.5a illustrates the amount of supervision on average used at each node of the derived hierarchies using different training strategies (See § 5.2.3). For reference, the amount of supervision required in the flat learning setting is illustrated. The amounts of supervision illustrated in the hierarchical learning settings are those required to attain a comparable accuracy with the flat learning setting. In addition, the amount of supervision is also assessed on (i) randomly picked hierarchies, (ii) the set of domain expert-defined hierarchies, and (iii) hierarchies derived using the approach defined in [OHA21b], which is based on the transfer-affinity between concepts to build the hierarchies. It is worth noticing that the hierarchies derived using our proposed approach achieve competitive performances while using far fewer training examples (approx. 2×10^{-3} examples) compared to the other hierarchies. This suggests that the concept grouping proposed by our measures reflects the actual concept dependence exhibited in the data. On the other hand, the need for supervision is more pronounced when using the regular training strategy.

5.3.3 Proposed measures and concept separability

Here we study the correlation between the proposed measures (cohesion and dispersion) and the separability of the grouped concepts. How do the measures of

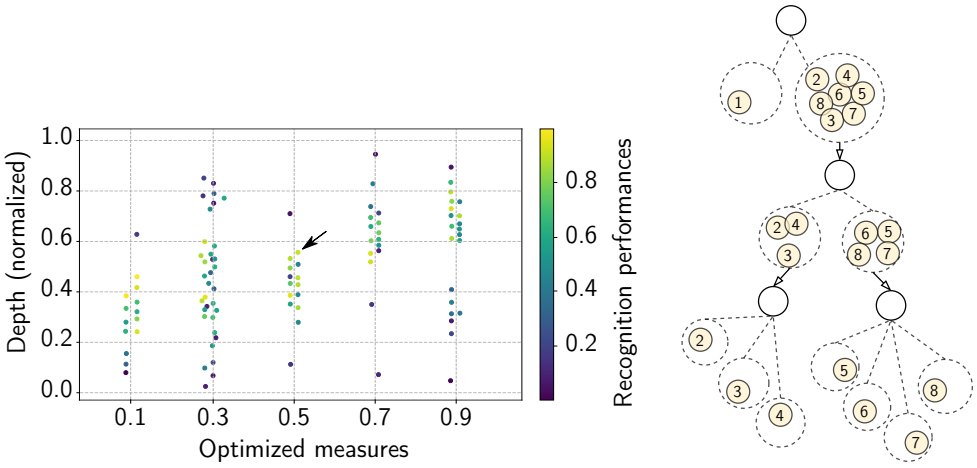


Figure 5.6: (left) Link between the proposed measures (x-axis) and the properties (depth and arity) of the derived hierarchies (y-axis). The final per-concept recognition performances are also depicted with varying colors. (right) One of the derived hierarchies corresponding to the arrow in the left.

cohesion and dispersion change when we go down the hierarchy? And above all, what is the impact of all this on the derived hierarchies? Are they deeper, i.e., are the best clustering solutions the ones that very quickly decompose the groups of concepts into atomic ones? Or, on the contrary, those trying to keep the concepts grouped until the leaves? How does this affect the learning of groups? How does this ultimately affect the recognition of atomic concepts? Which concepts really benefit from being grouped together? And, which concepts benefit from being rather learned on their own? We assess some of these questions here.

Figure 5.6, illustrates the link between the proposed measures and the properties of the derived hierarchies in terms of depth and arity along with the final per-concept recognition performances. We particularly focus on the effect of various scores of the cohesion and dispersion measures on the derived hierarchies and what does this imply in terms of concept grouping and how accurately the atomic concepts are recognized. In theory, optimal hierarchies would be those keeping the concepts grouped while going down the hierarchy, which results in deeper hierarchies in a way that the biases of groups are leveraged to a greater extent. Indeed, this is what we can see for high values of the optimized measures (≥ 0.8), where we get a fairly large number of deep hierarchies which are accompanied by fair recognition performances (approx. 70%). An increase in the computed measures

Hyperparam.	Groups of concepts [0][1-7]	[1,2,3][4-7]	[1][2][3]	[4][5][6][7]
First layer				
Kernel size	0.496	0.021	0.026	0.079
# of filters	0.325	0.078	0.014	0.124
Stride	0.852	0.745	0.752	0.664
Second layer				
Kernel size	0.147	0.578	0.454	0.125
# of filters	0.452	0.327	0.273	0.368
Stride	0.662	0.491	0.765	0.054
Third layer				
Kernel size	0.654	0.584	0.027	0.041
# of filters	0.076	0.025	0.581	0.031
Stride	0.324	0.558	0.754	0.017

Table 5.3: Hyperparameters' importance obtained through the fANOVA analysis of the hierarchy depicted in Figure 5.6.

results in a slight augmentation in the recognition performances globally.

5.3.4 Hyperparameters and inductive biases

The hierarchical structuring of the concepts allows us to circumscribe the search space for each group of concepts. The bias learned at each non-leaf node is consequently more adapted to each group. However, one question that remains unclear and could open room for further improvement is the link between these various biases. In other words, is there a way to go beyond and structure the biases such that a given learner can share them with its descendant in the hierarchy? Indeed, various works touched on this aspect from the operational point of view, such as [TMF07; ZXW11] which leveraged the transfer of orthogonal representation between children and parents in hierarchies and the second approach that we propose (see § 5.4), where we will use transfer-affinity between concepts and groups of concepts, but this time to simultaneously build the hierarchy of concepts.

An interesting way to tackle this question is related to the works around weight-agnostic neural architectures and those around the interpretation of the hyperparameters as inductive biases [LJ09; FC18; GH19]. Here, we provide a solution to investigate the link between the inductive biases used by the learners assigned to one of the derived hierarchies. For this, we design an experimental setting in which the architectures (hyperparameters) of the learners assigned to the non-

leaf nodes are optimized in a weight-agnostic fashion. This learning paradigm allows us to shift the focus from the set of weights toward the hyperparameters of the architectures. In a second step, we perform hyperparameter importance assessment following the methodology in [HHL14] and in one of our previous contributions [HO20] in order to check how inductive biases behave in the learned hierarchy of concepts.

Table 5.3 summarizes the obtained results from the hyperparameters assessment process. It illustrates the importance of each of the optimized hyperparameters at each node of the considered hierarchy. In particular, among the optimized hyperparameters that define the architecture of the learners assigned to the hierarchy, there are the *kernel size*, *number of filters*, and *stride* of convolution-base neural network layers. Their predefined ranges can be found in the code repository. It is worth noting the appearance, at each level of the hierarchy, of a specific set of hyperparameters that exhibit high importance as captured by the fANOVA framework. In particular, the *stride* of all three layers has the highest importance among this set. This hyperparameter determines the portion of the signal the convolution layers process at a time. The size of this portion is specific to each group of concepts, e.g., smaller for dynamic activities and bigger for static ones.

5.4 Concepts structuring based on transfer affinity

In this part, we propose a data-driven approach to structure the considered concepts in a bottom-up process instead of the top-down one presented above. Again, with the goal of maximizing transfer, sharing, and reuse across the constructed structures. This approach is based on transfer affinity to determine an optimal organization of the concepts. This is a powerful technique based on transfer learning, which showed interesting empirical properties in various domains [Zam+18; PRS19]. Our approach starts by computing concept dependencies that exist in the data domain using the transfer affinity scores. The closest concepts are then fused hierarchically with each other. When taking a bottom-up process, the complete hierarchy, including the parameters assigned to each non-leaf node, can be learned incrementally by reusing what was learned on the way, i.e., while computing the transfer affinity scores (see details in the following). We perform experiments to show the effectiveness of the proposed approach and a comprehensive comparative analysis of its various stages. This raises noticeably interesting questions about concept dependencies and the required amount of supervision. In the following, we detail the different parts of our approach, which are illustrated in Figure 5.7. Specifically, we introduce the three stages of our approach in detail: *Concept sim-*

ilarity analysis, *Hierarchy derivation*, and *Hierarchy refinement*.

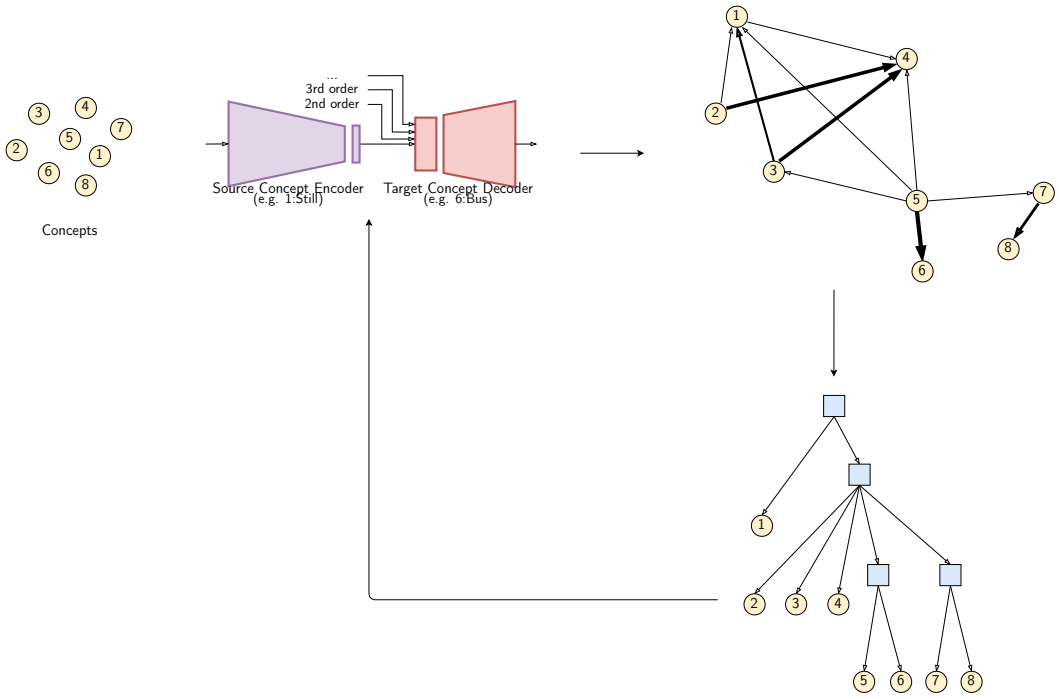


Figure 5.7: Our solution involves several repetitions of 3 main steps: (1) *Concept similarity analysis*: encoders are trained to output an appropriate representation for each source concept, which is then fine-tuned to serve target concepts. Affinity scores are depicted by the arrows between concepts (the thicker the arrow, the higher the affinity score). (2) *Hierarchy derivation*: based on the obtained affinity scores, a hierarchy is derived using the hierarchical agglomerate clustering approach. (3) *Hierarchy refinement*: each non-leaf node of the derived hierarchy is assigned with a model that encompasses an appropriate representation as well as additional dense layers which are optimized to separate the considered concepts.

5.4.1 Concept similarity analysis

In order to define concept similarity, we leverage two measures of similarities among concepts as transferability and dependency. Aside from the nice empirical properties of this measure, including quality, gains, and universality, it allows us to reuse what has been learned so far at the lower levels of the hierarchies. Indeed,

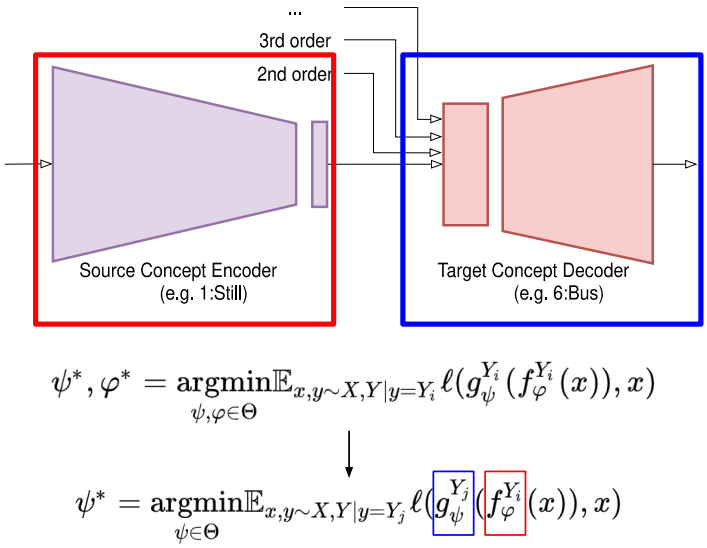


Figure 5.8: Concept similarity analysis: encoders are trained to output, for each source concept, an appropriate representation which is then fine-tuned to serve target concepts.

we leverage the models that we learned during this step and use them with a few additional adjustments in the final hierarchical learning setting.

Transfer-based affinity. Given the set of concepts Y, we compute during this step an affinity matrix that captures the notion of transferability and similarity among the concepts (see Figure 5.9).

For this, we first compute for each concept $Y_i \in Y$ an encoder $f_\varphi^{Y_i}$ (parameterized by φ) that learns to map the Y_i labeled inputs, to a latent vector as z_{c_i}. Learning the encoder's parameters consists in minimizing the reconstruction error, satisfying the following objective function [Vin+10]:

$$\psi^*, \varphi^* = \underset{\psi,\varphi\in\Theta}{\text{argmin}}\,\mathbb{E}_{x,y\sim X,Y|y=Y_i}\ell(g_\psi^{Y_i}(f_\varphi^{Y_i}(x)), x), \tag{5.6}$$

where g_ψ^y is a decoder (parameterized by ψ) that maps back the learned representation into the original input space. We propose to leverage the learned encoder for a given concept Y_i, to compute the affinities in comparison to other concepts via fine-tuning of the learned representation. Precisely, we fine-tune the encoder $f_\varphi^{Y_i}$ to account for a target concept $Y_j \in Y$, $Y_i \neq Y_j$. This process consists, similarly, of minimizing the reconstruction error, however rather than using the decoder $g_\psi^{Y_i}$ learned above, we design a genuine decoder $g_\psi^{Y_j}$ that we learn from scratch. The

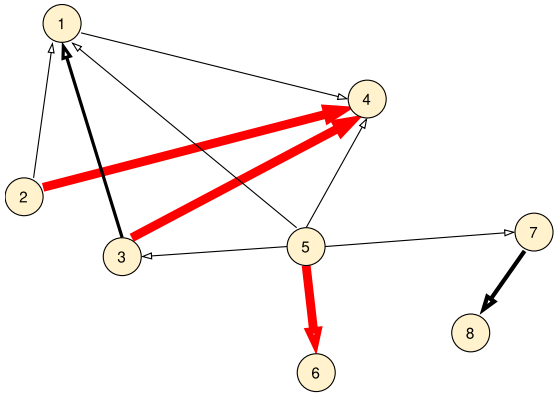


Figure 5.9: Example of the obtained similarity scores using the SHL dataset. The arrows in red have the higher similarity scores.

corresponding objective function is

$$\psi^* = \underset{\psi \in \Theta}{\operatorname{argmin}}\, \mathbb{E}_{x,y \sim X,Y|y=Y_j} \ell(g_\psi^{Y_j}(f_\varphi^{Y_i}(x)), x), \tag{5.7}$$

where the parameters of the encoder φ are frozen, i.e., only parameters of the genuine decoder ψ for the target concept Y_j are updated during this process.

We use the performance of this process as a *similarity score* from Y_i to Y_j which we denote by $p_{Y_i \to Y_j} \in [0, 1]$. We refer to the number of examples belonging to the concept Y_j used during fine-tuning as the *supervision budget*, denoted as b, which is used to index a given measure of similarity. It allows us to have an additional indicator as to the similarity between the considered concepts. Finally, the *affinity similarity* score is computed as $\frac{\alpha \cdot p_{Y_i \to Y_j} + \beta \cdot b}{\alpha + \beta}$. We set α and β to be equal to $\frac{1}{2}$. Figure 5.9, provides an example of the obtained similarity scores using the SHL dataset.

Properties. In many applications, e.g., computer-vision [Zam+18] and natural language processing [PRS19], several variants of the transfer-based similarity measure have been shown empirically to improve three aspects: (i) the *quality* of transferred models (wins against fully supervised models), (ii) the *gains*, i.e., win rate against a network trained from scratch using the same training data as transfer networks, and more importantly (iii) the *universality* of the resulting structure. Indeed, the affinities based on transferability are stable despite the variations of a big corpus of hyperparameters. We provide similarly some empirical evidence (Section 5.5.2) for the appropriateness of the transfer-based affinity measure for

the separability of similar concepts and the difficulty of separating concepts that exhibit low similarity scores.

5.4.2 Hierarchy derivation

Given the set of *affinity scores* obtained previously, we derive the most appropriate hierarchy, following an agglomerative clustering method combined with some additional constraints (see Figure 5.10). The agglomerative clustering method proceeds by a series of successive fusions of the concepts into groups and results in a structure represented by a two-dimensional diagram known as a dendrogram. It works by (1) forming groups of concepts that are close enough and (2) updating the affinity scores based on the newly formed groups. This process is defined by the recurrence formula proposed by [LW67]. It defines a distance between a group of concepts (k) and a group formed by fusing i and j groups (ij) as $d_{k(ij)} = \alpha_i d_{ki} + \alpha_j d_{kj} + \beta d_{ij} + \gamma |d_{ki} - d_{kj}|$, where d_{ij} is the distance between two groups of concepts i and j. By varying the parameter values $\alpha_i, \alpha_j, \beta$, and γ, we expect to get clustering schemes with various characteristics.

In addition to the above updating process, we propose additional constraints to refine further the hierarchy derivation stage. Given the dendrogram produced by the agglomerative method above, we define an *affinity threshold* δ such that if the distance at a given node is $d_{ij} \geq \delta$, then we merge the nodes to form a unique subtree. In addition, as we keep track of the quantities of data used to train and fine-tune the encoders during the transfer-based affinity analysis stage, this indicator is exploited to inform us as to which nodes to merge. Let $\mathcal{T}$ be the derived hierarchy (tree), and let t index the non-leaf or internal nodes. The leaves of the hierarchy correspond to the considered concepts. For any non-leaf node t, we associate a model θ_t encompassing (1) an encoder obtained using the previously described process and (2) additional dense layers (on top of the encoder) for classification that output decision boundaries based on the representations produced by the encoder (see Figure 5.11).

5.4.3 Hierarchy refinement

After explaining the hierarchy derivation process, we will discuss (1) which representations are used in each individual model; and (2) how each individual model (including the representation and the additional dense layers) is adjusted to account for both the local errors and also those of the hierarchy as a whole.

Which representations to use? The question discussed here is related to the encoders to be used in each non-leaf node. For any non-leaf node t we distinguish two cases: (i) all its children are leafs; (ii) it has at least one non-leaf node. In

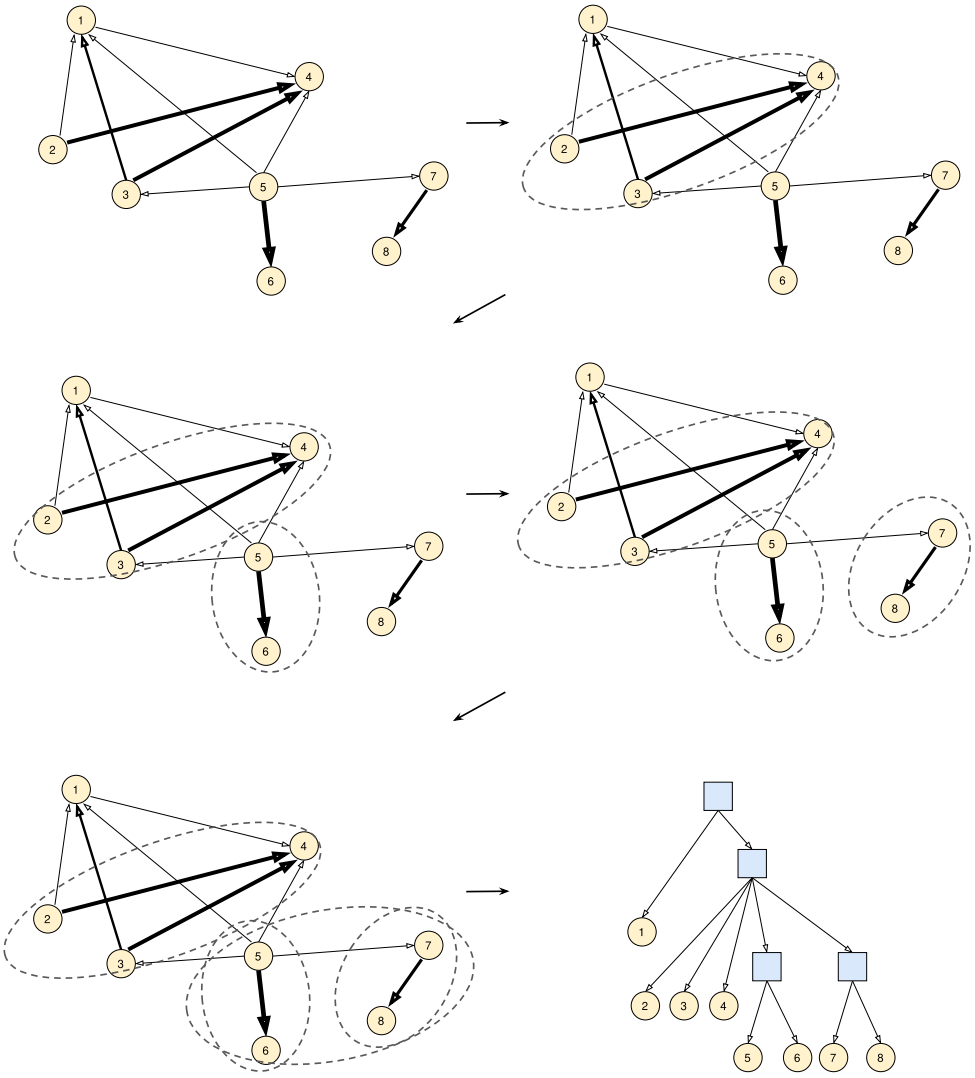


Figure 5.10: An example of the hierarchy derivation process applied to the SHL. The steps of the derivation process are depicted with circles that group each time a subset of the considered concepts.

the first case, the final considered representation, associated with the non-leaf node, is the representation learned in the concept affinity analysis step (first-order transfer-based affinity) (see Figure 5.11). In the second case, we can either fuse the nodes (for example, in a case of classification between 3 concepts, we get all 3 together rather than, first {1} vs. {2,3}, then {2} vs. {3}) or keep them as they are and leverage the affinities based on higher-order transfer where, rather

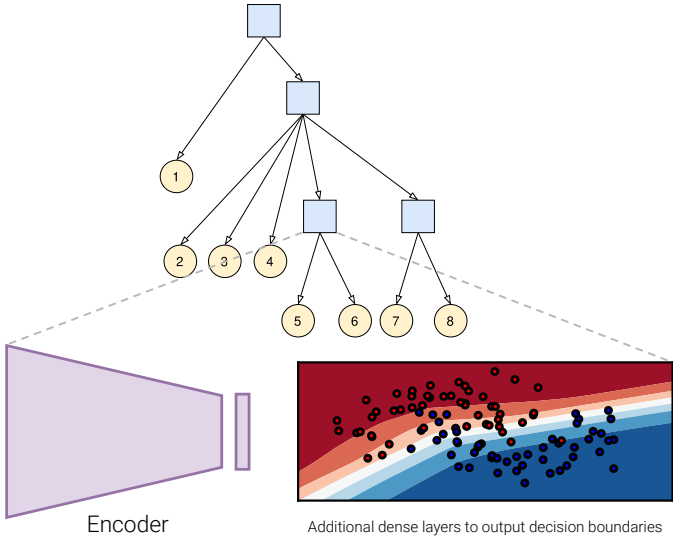


Figure 5.11: For any non-leaf node of the derived hierarchy, we associate a learning model that encompasses an encoder that maps inputs to their corresponding latent representations and additional dense layers that output decision boundaries based on the representations produced by the encoder.

than accounting for a unique target concept, the representation is then fine-tuned. Figure 5.12 illustrates how transfers are performed between non-leaf nodes models. We index the models with the encoder $\theta_{[Y_i]}$. In the case of higher-order transfer, the models are indexed using all concepts involved in the transfer, i.e., $\theta_{[Y_{i,j,\ldots}]}$.

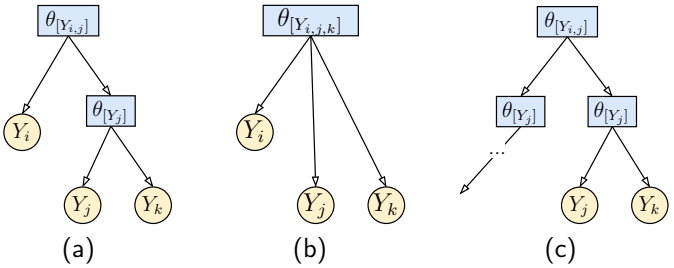


Figure 5.12: The transfer is performed between non-leaf node models. The hierarchy in (a) can be kept as it is merged to form the hierarchy in (b). (b): a high-order transfer between the concepts Y_i, Y_j, and Y_k is performed. (c): no transfer can be made.

Adjusting weights of the models. Given the encoder f_φ assigned to any non-leaf node t, the additional dense layers ω parameterizing the function g are trained to output a hypothesis based on the most appropriate representations learned earlier. The parameters of the encoder are frozen while the additional dense layers are trained as

$$\underset{\omega \in \Theta}{\operatorname{argmin}} \, \mathbb{E}_{x,y \sim X,Y|z=f_\varphi(x)} \ell(g_\omega(z), y). \tag{5.8}$$

The parameters are adjusted to account for local errors as well as for global errors related to the hierarchy as a whole.

5.5 Experiments

Empirical evaluation of this second approach is performed following three steps: we evaluate classification performances in the hierarchical setting (Section 5.5.1); then, we evaluate the transfer-based affinity analysis step and the properties related to the separability of the considered concepts (Section 5.5.2); finally, we evaluate the derived hierarchies in terms of stability, performance, and agreement with their counterparts defined by domain experts (Section 5.5.3). A similar experimental setup as the previous experimental part (Section 5.3.1) is used in the following.

Implementation details. We use Tensorflow for building the encoders/decoders. We construct encoders by stacking Conv1d/ReLU/MaxPool blocks. These blocks are followed by a Fully-Connected/ReLU layers. The encoder's performance estimation is based on the validation loss and is framed as a sequence classification problem. As a preprocessing step, annotated input streams from the SHL dataset are segmented into sequences of 6000 samples which correspond to a duration of 1 min. given a sampling rate of 100 Hz. For weight optimization, we use stochastic gradient descent with a Nesterov momentum of 0.9 and a learning rate of 0.1 for a minimum of 12 epochs (we stop training if there is no improvement). Weight decay is set to 10^{-4}. Furthermore, to make the neural networks more stable, we use batch normalization on top of each convolutional layer.

5.5.1 Evaluation of the hierarchical classification performances

In these experiments, we evaluate the flat classification setting using neural networks which constitute our baseline for the rest of the empirical evaluations. To compare our baseline with the hierarchical models, we make sure to get the same

Algorithm 8: Hierarchical learning of dependent concepts based on transfer affinity. Summary of the concept structuring process starting from an empty tree ($\mathcal{T} = \emptyset$) until a full hierarchy with trained models ($\mathcal{T}_{\theta_1^*,\ldots,\theta_T^*}$) is returned.

Input : (i) $X = \{(x_i, y_i)\}_{i=1}^{N}$ set of annotated training examples;
(ii) $Y = \{Y_1, \ldots, Y_n\}$ denotes the set of concepts.

begin
  $\mathcal{T} \leftarrow \emptyset$
  $i \leftarrow 1$
  $B \leftarrow$ subset of X as a supervision budget ; % $|B| = b$.
  repeat
    Compute i^{th} order concept affinity (Sect. 5.4.1)
    ; % *i.e., for the combinations of* $i+1$ *concepts among the set of concepts*
    ; % *Initialize the weights of the learners being assigned to the hierarchy*
    if $\mathcal{T}$ *is empty* **then**
      Derive the concept hierarchy $\mathcal{T}$ (Sect. 5.4.2)
      ; % *select the pairs of encoder-decoder with the highest affinity scores*
      ; % *given the supervision budget, fine-tune the encoders to account for the target concept*
      ; % *Initialize the weights of the learners assigned to the hierarchy*
      $\mathcal{T}_{\theta_1,\ldots,\theta_T} \leftarrow$ `initialize`()
    else
      Update the hierarchy $\mathcal{T}$ (Section 5.4.3)
    end
    ; % *Update the weights of the models (Algorithm 6)*
    $\mathcal{T}_{\theta_1,\ldots,\theta_T} \leftarrow$ `trainHierarchy`$(B, \mathcal{T})$
    ; % *Evaluate hierarchy globally*
    error $\leftarrow$ `evaluate`$(X, \mathcal{T})$
    $i \leftarrow i + 1$; % *increase affinity order*
    ; % *increase supervision budget, where* $\hat{B}$ *is a subset of* X.
    $B \leftarrow B \cup \hat{B}$
  until *convergence (error* $< \epsilon$*)*;
end

Result: Trained hierarchy $\mathcal{T}_{\theta_1^*,\ldots,\theta_T^*}$

complexity, i.e. comparable number of parameters as the largest hierarchies including the weights of the encoders and those of the additional dense layers. We also use Bayesian optimization based on Gaussian processes as surrogate models to select the optimal hyperparameters of the baseline model [SLA12; HOA20b].

Per-node performances. Figure 5.13 shows the resulting per-node performances, i.e. how accurately the models associated with the non-leaf nodes can predict the correct subcategory averaged over the entire derived hierarchies. The nodes are ranked according to the obtained per-node performance (top 10 nodes are shown) and accompanied by their appearance frequency. It is worth noticing that the concept 1:*still* learned alone against the rest of the concepts (first bar) achieves the highest gains in terms of recognition performances while the appearance frequency of this learning configuration is high (more than 60 times). We see also that the concepts 4:*bike*, 5:*car*, and 6:*bus* grouped together (5th bar) occur very often in the derived hierarchies (80 times) which is accompanied by fairly significant performance gains ($5.09 \pm 0.3\%$). At the same time, as expected, we see that the appearance frequency gets into a plateau starting from the 6th bar (which lasts after the 10th bar). This suggests that the most influential nodes are often exhibited by our approach.

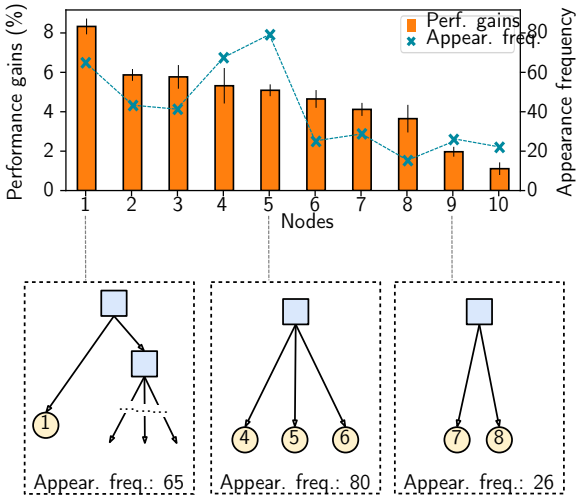


Figure 5.13: Per-node performance gains averaged over the entire derived architectures (similar nodes are grouped, and their performances are averaged). The appearance frequency of the nodes is also illustrated. Each bar represents the gained accuracy of each node in our hierarchical approach. For example, the 8th bar corresponds to the concepts 2:*walk*-3:*run*-4:*bike* grouped together.

Per-concept performances. We further ensure that the performance improvements we get at the node levels are reflected at the concept level. Experimental

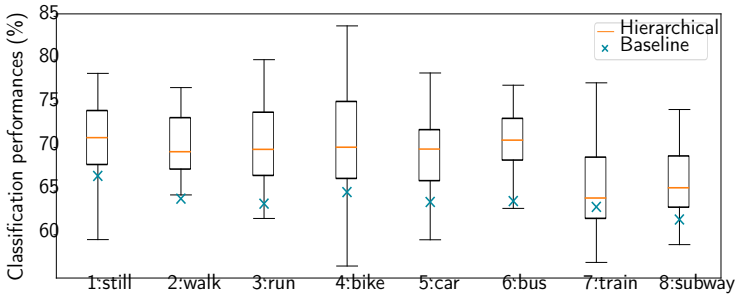


Figure 5.14: Recognition performances of each individual concept, averaged over the entire derived hierarchies. For reference, the recognition performances of the baseline model are also illustrated.

results show the recognition performances of each concept, averaged over the whole hierarchies derived using our proposed approach. We indeed observe that there are significant improvements for each individual concept over the baseline (flat classification setting). We observe that again 1:*still* has the highest classification rate ($72.32 \pm 3.45\%$) and an improvement of 5 points over the baseline. Concept 6:*bus* also exhibits a roughly similar trend. On the other hand, concept 7:*train* has the least gains ($64.43 \pm 4.45\%$) with no significant improvement over the baseline. Concept 8:*subway* exhibits the same behavior suggesting that there are undesirable effects that stem from the definition of these two concepts.

5.5.2 Evaluation of the affinity analysis stage

These experiments evaluate the proposed transfer-based affinity measure. We assess, the separability of the concepts depending on their similarity score (for both the transfer-affinity and supervision budget) and the learned representation.

Appropriateness of the transfer-based affinity measure. We reviewed above the nice properties of the transfer-based measure, especially the universality and stability of the resulting affinity structure. The question that arises is related to the separability of the concepts that are grouped together. Are the obtained representations, are optimal for the final models used for the classification? This is what we investigate here. Figure 5.15b shows the decision boundaries generated by the considered models, which are provided with the learned representations of two concepts. The first case (top right) exhibits a low-affinity score, and the second case (bottom right) shows a high-affinity score. In the first case, the boundaries are unable to separate the two concepts while it gets a fairly distinct frontier.

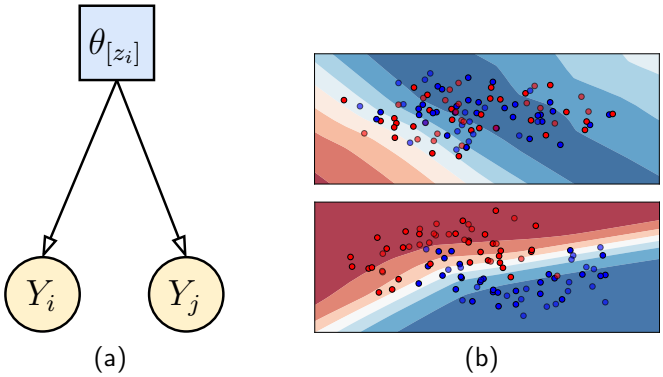


Figure 5.15: (a) Non-leaf node grouping concepts Y_i and Y_j. (b) Decision boundaries generated by the additional dense layers plugged into the non-leaf node using an encoder (representation) fine-tuned to account for (top) the case where Y_i and Y_j are dissimilar (low-affinity score) and (bottom) the case where Y_i and Y_j are similar (high-affinity score).

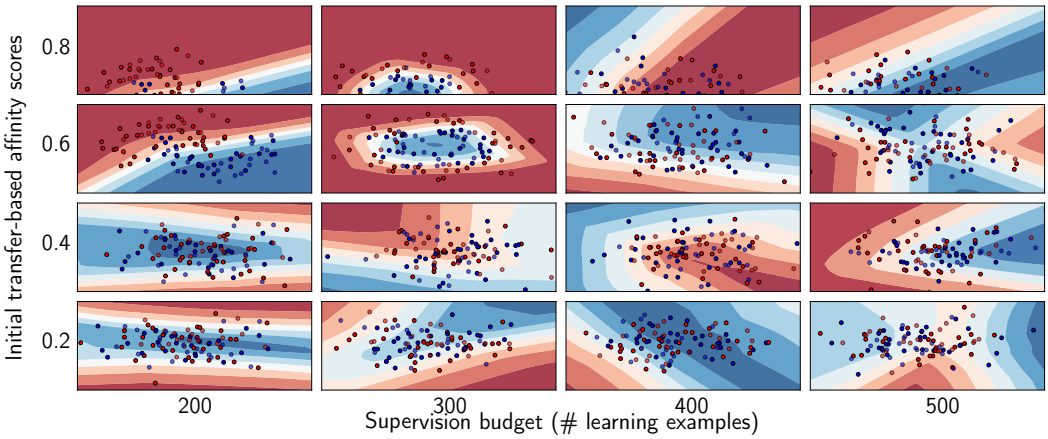


Figure 5.16: Decision boundaries obtained by the additional dense layers trained on the representations z_t as a function of the distance between the concepts (y-axis) and the supervision budget (x-axis).

Impact on the models' decision boundaries. We train different models with various learned representations in order to investigate the effect of the initial affinities (obtained solely with a set of 100 learning examples) and the supervision budget (additional learning examples used to fine-tune the obtained representation) on the classification performances of the models associated with the non-leaf nodes of our hierarchies. Figure 5.16a shows the decision boundaries generated by various

models as a function of the distance between the concepts (y-axis) and the supervision budget (x-axis). Increasing the supervision budget to some larger extents (more than $\sim$ 300 examples) results in a substantial decrease in the classification performances of the models. This suggests that, although our initial affinity scores are decisive (e.g. 0.8), the supervision budget is tightly linked to generalization. This shows that a trade-off (controlled by the supervision budget) between separability and initial affinities arises when we seek to group concepts together. In other words, the important question is whether to increase the supervision budget indefinitely (in the limits of available learning examples) in order to find the most appropriate concepts to fuse with, while expecting good separability.

5.5.3 Universality and stability

We demonstrated in the previous section the appropriateness of the transfer-based affinity measure to provide distance between concepts as well as the existence of a trade-off between concepts separability and their initial affinities. Here we evaluate the **universality** of the derived hierarchies as well as their **stability** during adaptation with respect to our hyperparameters (affinity threshold and supervision budget). We compare the derived hierarchies with their domain experts-defined counterparts, as well as those obtained via a random sampling process. Figure 5.17 shows some of the hierarchies defined by the domain experts (first row) and sampled using the random sampling process. For example, the hierarchy depicted in Figure 5.17d corresponds to a split between static (1:*still*, 5:*car*, 6:*bus*, 7:*train*, 8:*subway*) and dynamic (2:*walk*, 3:*run*, 4:*bike*) activities. The difference between the hierarchies depicted in Figure 5.17a and 5.17b is related to 4:*bike* activity which is linked first to 2:*walk* and 3:*run* then to 5:*car* and 6:*bus*. A possible interpretation is that in the first case, biking is considered as "on feet" activity, while in the second case as "on wheels" activity. What we observed is that the derived hierarchies tend to converge towards the expert-defined ones.

Method	Agree.	perf. avg.± std.
Expertise	-	72.32±0.17
Random	0.32	48.17±5.76
Proposed	0.77	75.92±1.13

Table 5.4: Summary of the recognition performances obtained with our proposed approach compared to randomly sampled and expert-defined hierarchies.

We compare the derived hierarchies in terms of their level of agreement. We use for this assessment, the Cohen's kappa coefficient [Coh60] which measures

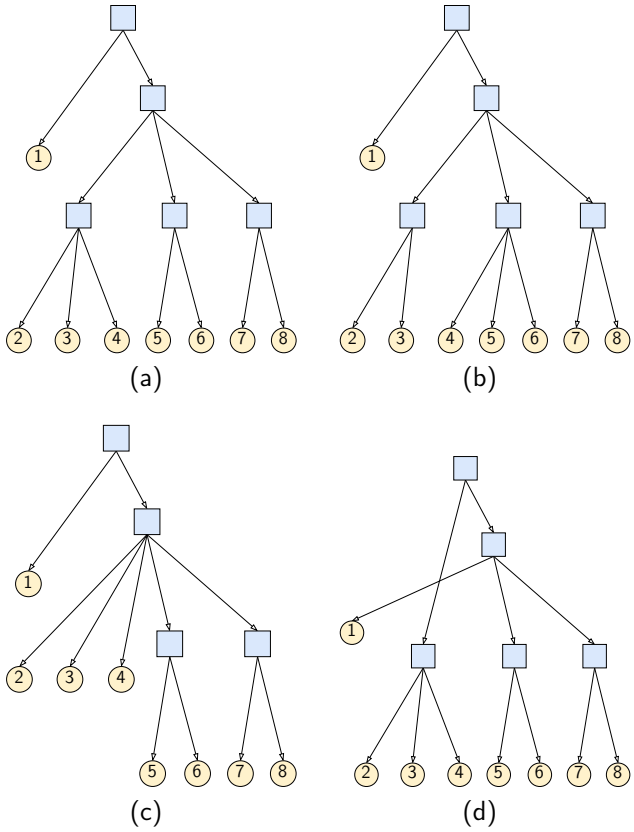


Figure 5.17: Examples of hierarchies: (a) defined via domain expertise, (b-c) derived using our approach, and (d) randomly sampled. Concepts 1—8 from left to right.

the agreement between two raters. The first column of Table 5.4 provides the obtained coefficients. We also compare the average recognition performance of the derived hierarchies (second column of Table 5.4). In terms of stability, as we vary the design choices (hyperparameters) defined in our approach, we found that the affinity threshold has a substantial impact on our results with many adjustments involved (12 hierarchy adjustments on avg.), whereas the supervision budget has a slight effect, which confirms the observations in Sec. 5.5.2.

5.6 Conclusion

We explored in this chapter different approaches for guiding the learning process by structuring the concepts to learn into hierarchies. We have proposed two approaches: one corresponding to a top-down strategy based on the clustering

and the decomposition of concept groups (§ 5.2); and another corresponding to a bottom-up strategy based on the calculation of transfer-affinity between concepts followed by the composition of concepts starting from those exhibiting a weak affinity to the transfer until the strongest (§ 5.4).

The former approach starts by determining a suitable structure for the concepts according to a transfer affinity-based measure. The latter approach starts by clustering groups of atomic concepts close enough to be learned together using cohesion and dispersion measures. The clustering approach substantially reduces the number of tree candidates for grouping the atomic concepts. Afterward, optimal representations and classifiers are characterized, which are then refined to account for both local and global errors.

The two approaches are distinguished by the fact that learning in the top-down approach is done in two stages, unlike the bottom-up approach. Indeed, in the top-down approach, the models are assigned to the nodes of the hierarchy only when the construction phase of the hierarchy is completed. Whereas in the bottom-up approach, the learning is done in parallel because the transfer models are learned at the same time as the groups of concepts are formed, based moreover on the transfer affinity calculated from these models.

Empirical evaluations demonstrated superior results using the hierarchies derived using our proposed approaches on a dataset collected in real-life settings, which is susceptible to concept overlaps (in addition to the intrinsic multi-inheritance of the featured concepts). The proposed approaches allow us to reduce the exponential theoretical complexity of basic hierarchical learning settings drastically. As we started to analyze and discuss in § 5.3.4 and as explored in some works, such as [TMF07; ZXW11], the inductive biases learned at each node of the hierarchy can be exhibited and leveraged in a way that some aspects will no longer require to be learned again from scratch. We provide theoretical bounds for the problem and empirically show that using our approach, we are able to improve the performances and robustness of activity recognition models over a flat classification baseline.

Moreover, from a purely operational point of view, the hierarchical learning of concepts would, among other things, make it possible to implement collaborative and decentralized processing mechanisms. For example, learning sub-problems at the higher levels of the hierarchies can be (pre-)processed at the levels of the final extremities of the system. The more specific sub-problems require heavier processing and, therefore, more substantial infrastructures. The (pre-)processing that is carried out at the levels of the final extremities of the system serves to facilitate learning at the central and intermediate levels. This can be enabled by mechanisms for transferring and sharing (or structuring) inductive biases.

Even if the hierarchical structuring of the concepts allows us to circumscribe the search space for each group of concepts and consequently get inductive biases

that are more adapted to each group, the proposed model can be further improved to get even better results on the final atomic concepts while using less supervision. In addition to supporting the necessity of organizing concept learning, our experiments raise interesting questions for future work. Noticeably, Sec. 5.5.2 asks what is the optimal amount of supervision for deriving the hierarchies. Future work follows various axes: (1) Explore other structures other than hierarchies, including the study of different approaches for searching and exploring the search space of different hierarchical types, noticeably lattices; (2) Explore strategies that do not rely on fixing any type of structure and consider it as a parameter that can be learned. Some works in this sense are [KT08; Ten+11]. Ultimately, one goal is to make the whole process trainable in an end-to-end fashion, which involves formulating the clustering and hierarchy derivation steps in a continuous relaxation scheme; (3) Express the transformations underlying the various groups of concepts via invariants and enforce them while training the different learners assigned to the structure. This is what we started to explore in § 5.3.4, where we assessed how inductive biases are inherited from one node to another; (4) Building on the way inductive biases are shared from one node to another, future works include leveraging the parameter-sharing scheme, introduced in [RSP17], to ensure in a principled way that the domain invariants are reproduced in the final learned model.

We saw in this chapter how to leverage the semantics of the label space to organize (or structure) the learning process. Hierarchical structures were explored in this sense, and the learning process benefited from the proposed transfer and sharing mechanisms across the levels of the hierarchies.

Chapter 6

Abstracting the context and modeling data relativity

In this chapter, we focus on the collaborative aspects of the massively distributed sensing nodes and the ways the conciliation of decentralized learners can be improved. We investigate approaches that can efficiently fuse the relative views provided by the sensing environments, abstract them from their contextual bias, and conciliate the decisions taken by decentralized learners while considering their relativity. The content of this chapter is based on [OH22] and [HO22].

We have reviewed, in the two previous chapters, various strategies for the integration of structural constraints in the learning process. Some focused on the input data (inputs), and others on the concepts to learn (outputs). The main advantage of such strategies lies in the possibility of having finer control over the learning process.

In this chapter, we will approach the abstraction of the heterogeneity induced by various effects, in particular those related to the relative positions of the data sources as well as to the contexts which surround them. The idea is to be able to identify and isolate the components linked to the effects of heterogeneity. Similarly to the previous chapters, we seek to put the learner in a good position to learn. Indeed, the strategies that we develop in this chapter aim to learn (in the sense of bias learning) transformations that project the data into spaces abstracted from various biases (those linked to the position of the data generators in particular).

In IoT applications, data generated from different sensing devices and locations are embodied with varying contexts. The devices offer specific perspectives on the problem of interest depending on their location, e.g., on-body sensor deployment for activity recognition. Furthermore, in applications involving moving targets

encompassing different parts, such as human activity recognition from on-body sensor deployments, the sensing devices are tightly linked with the dynamics of the parts of the target they are located on [BBS14; Shi+20; BY20; HO20; HO21b; Car+18; MSB20]. As a consequence, the movements of the area on which the sensing devices are positioned generate data of two different but complementary natures. For instance, in Figure 6.1, the data of the movement collected from the hand sensors combines data of the whole body intertwined with data related to the movement of the hand in relation to the body. The first concerns the movement of the position relative to the target itself, and the second concerns the movement of the target relative to its surroundings. In the case of human activity recognition, we notice, for example, that the kinetics of the hand movements during a race can be decomposed into a circular movement (CM) of the hand relative to the shoulder and a translation movement (TM) associated with the whole body [MSB20]. At least three practical implications can be devised from this: (i) CM data are enough to learn some target concepts, e.g., the hand kinetics movement is enough to determine if a person is at rest or running; (ii) CM data from different positions, e.g., hand and torso, cannot be shared and mixed together. Otherwise, this generates noise and confusion during the learning process; (iii) only TM data can be shared among the different positions as these data are of the exact same nature but taken from different points of view (positions or perspectives).

For this, we will draw inspiration from two key principles: (1) abstraction in parameters and meta-parameters (feature learning and reuse) where the principle is to be able to capture the heterogeneous components with the parameters and the common components in the meta-parameters; and (2) the exploitation of similarities between tasks where the idea is (i) to exhibit different forms of structures (e.g., symmetries), (ii) to represent them with adapted tools (e.g., special Euclidean group) and (iii) to incorporate them into the learning process. The idea is to capture the context of each data source in specific components of the learning model. This way of proceeding will therefore have the additional consequence of facilitating model adaptation since the components that evolve because of the context, in reality, are isolated and controlled. Remember that the context here does not simply refer to the position of the data sources but can encompass both the specificities of the data sources, their positions relative to the phenomenon of interest, their position relative to other data sources, etc.

We begin this chapter by describing structured environments and the basic federated learning approaches used in these types of environments. Also, we will discuss the impact of context on the learning process and present the formalization of the position abstraction problem as a latent representation learning problem (Section **??**). First, we investigate in Section 6.2 the emergence of such "compartmentalizations" into parameters and meta-parameters (mentioned previously in

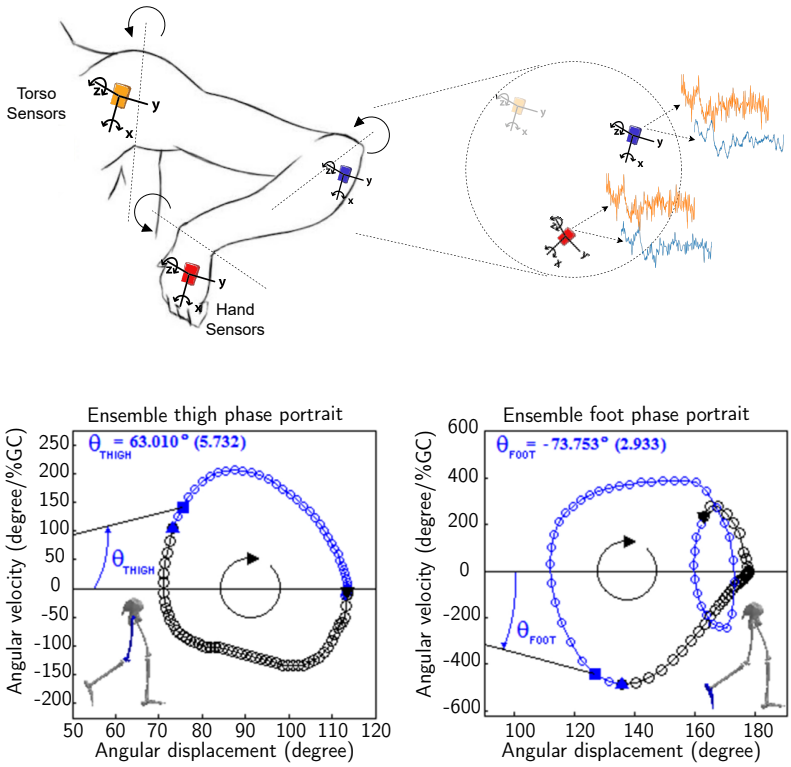


Figure 6.1: (left) The hand sensor undergoes two types of movements. One is of the same nature as the torso and linked to the translational movement of the body. The other is linked to the movement of the hand locally relative to the body. (right) Phase plan showing the dynamics of the thigh and foot during gait cycle (GC) (⊸ 1%GC) extracted from the biomechanics works of [Car+18].

Section 2.4). In particular, using the so-called "disentanglement" approaches. In this part, we leverage the data decomposition into universal and position-specific components to improve activity recognition models. These components have distinctive contributions concerning the target concepts to learn. This brings an interesting property that allows us to fuse the universal components as seen from different points of view (positions) while identifying the position-specific components, which could serve as additional knowledge in situations when the position-specific components are not sufficient to recognize an activity. Without this data decomposition process, the local part of the data adds noise, which is challenging to manage when relying solely on standard aggregation techniques. Indeed, to integrate data from different positions (or clients), it is necessary to separate the data of the same nature (shareable) from the pure local ones linked to the

specific kinetics of the position. Similar data can and should be shared to improve recognition rates. However, the specific data must be processed locally, otherwise impacting the learning process.

In Section 6.4, we propose to exploit the structure of data source deployments, in particular their relativity. We leverage in this part the recent advances in machine learning literature, e.g., [VI20], which seek to integrate invariants in the form of, e.g., symmetries within the phenomena of interest, into the learning process. Symmetry is one of the invariants that is leveraged for its powerful properties and its promising ability to drastically reduce the problem size [CGF19; Fin+20; QBC20] by requiring fewer training examples than standard approaches for achieving the same performance. Group theory provides a useful tool for reasoning about invariance and equivariance. In particular, we introduce the notion of relativity between data generators and model it via the special Euclidean group, denoted by $SE(3)$, which encompasses arbitrary combinations of translations and rotations. The relative contribution of a data generator in the description of the phenomena of interest is expressed using elements of this group and used to constrain the separation process. This allows us to leverage further the notion of sharing, which is reflected in the conciliation process of the decentralized learning setting by promising improvements.

6.1 Problem formulation

Here we motivate and formalize the problem of context abstraction and relativity modeling to improve collaboration in massively distributed sensing environments. We first study the impact that the context surrounding the distributed data sources imposes on the learning process. In particular, the position bias that taints the data generated, for example, in distributed sensing deployments. The problem of abstraction is looked at from the lenses of data disentanglement. Domain knowledge about the geometrical relativity of data sources is also motivated as a means to further improve collaboration.

6.1.1 Setting

We consider settings where a collection $\mathcal{S}$ of M sensors (also called data sources), denoted $\{s_1, \dots, s_M\}$, are positioned respectively at positions $\{p_1, \dots, p_M\}$ on the object of interest, e.g., human body. Each sensor s_i generates a stream $\mathbf{x}^i = (x_1^i, x_2^i, \dots)$ of observations of a certain modality like acceleration, gravity, or video, distributed according to an unknown generative process. Furthermore, each observation can be composed of channels, e.g., three axes of an accelerometer.

In the case of the SHL dataset, the goal is to continuously recognize a set of human activity target concepts Y like running or biking. Data are generated from 4 smartphones, carried simultaneously at (hand, torso, hips, and bag body locations (see detailed description in Section 4.3). Sensors distributed in various positions of the space provide rich perspectives and contribute in different ways to the learning process, and decentralization has the potential to offer better generalization performances.

6.1.2 Sensing deployments and impact of the context

Long lines of research studied the impact of the varying contexts on machine learning algorithms and showed their fragility to viewpoint variations, e.g., [Hsi+20] in an FL setting. For example, basic convolutional networks are found to fail when presented with out-of-distribution category-viewpoint combinations, i.e., combinations not seen during training [Mad+20]. Similarly, in activity recognition, the diversity of users, their specific ways of performing activities, and the varying characteristics of the sensing devices have a substantial impact on performances [Sti+15; HO21b]. In these cases, the conditional distributions may vary across clients even if the label distribution is shared [Kai+19]. In decentralized approaches, several theoretical analyses bound this drift by assuming bounded gradients [YYZ19], viewing it as additional noise [KMR20], or assuming that the client optima are ϵ-close [Li+20a].

The impact of varying contexts is not limited to a skewed distribution of labels but is rather predominantly related to the aspects of the phenomenon being captured by the sensing devices depending on their intrinsic characteristics and locations. Depending on their disposition w.r.t. to the phenomena of interest, the sensing devices generate different views of the same problem. The diversity brought by these configurations in terms of views is beneficial but must be explicitly handled. Reconciling the various perspectives offered by these deployments using decentralized learning approaches requires several relaxations limiting their potential capabilities when the impact of the context on the data generation process is essential. Indeed, how to reconcile these different points of view, which can potentially be redundant or even seemingly contradictory to each other? When additional knowledge is available about the structure of the sensing environment, these challenges can be handled efficiently.

Traditional HAR approaches [BBS14; OR16; Yao+17] often consider the sensory inputs to be flattened therefore disregarding the significant impact of the various positional biases. Some approaches consider these problems from the perspective of deployment optimization, mainly focusing on the study of the optimal on-body sensors placement and its impact on the recognition of target activities [Ban+14; Shi+20; BY20]. There are also rare approaches offering pipelines

that include recognition of the position of the data generator followed by the activity recognition [YW16] or including an explicit model of the context [Eha+20; Asi+20]. Other approaches, e.g., [KL08], try to develop heuristics to improve the robustness of activity recognition models to sensor displacements. Regardless of the devised techniques, these approaches rely on centralized processing of the data, which does not match the intrinsic complementary nature of the data, thus limiting their potential capacities.

6.1.3 Abstraction of the position

As described previously, each sensor produces two types of orthogonal data. This problem can be formally defined as the construction of a factorized representation z being a composition of (i) position-invariant (abstract or universal) components vector z_A, and (ii) a position-specific (local) components vector z_P. On the one hand, the position-invariant components vector captures the features that are shared across all positions. On the other hand, the position-specific components vector captures specific and complementary insights with regard to the target concepts. The first problem to solve in our model is to build this data decomposition process for each sensor automatically.

Thanks to this process, for each sensor s_i located in position p_i, the associated local model will have the ability to disentangle the data interlaced between the local and universal components by projecting them into two separate representations z_A^i and z_P^i (or simply z_{p_i}). Components z_P will be useful only in a local model, while z_A can be used in the local model or shared with the same kind of components originating from all other models in a global model (or central server). This process has the potential to allow fine-grained control of the inference process. Indeed, one can leverage different configurations in order to get optimal recognition performances, while traditional HAR approaches often consider the inputs to be flattened and disregard the bias related to the position. We notice, for example, that in certain situations, the position-specific components alone are enough to recognize the activity, e.g., the circular movements of the hand are enough to distinguish between running and walking. In addition, since only position-independent data is shared, this process greatly reduces data heterogeneity. It, therefore, improves data aggregation techniques for clients in federated learning settings by sharing only the position-invariant data. When the data are not decomposed, the position-specific part of the data represents noise for the global system.

To deal with these two challenging complementary representations, we propose a model based on multi-level processing to abstract the position as described below. In this model, we suppose that the position-invariant components share the data with the central learner.

6.1.4 Relativity of viewpoints in structured sensing environments

Very often, knowledge about the relative geometry of the sensing devices and domain models describing the dynamics of the phenomena is available and can be leveraged and incorporated into the learning process. For example, the spatial structure of the sensors deployment and the induced views, sensors capabilities, and the perspectives (views) through which the data is collected (sensing model, range, coverage, position in space, position on the body, and type of captured modality) [AC09; WKA10; HOA20a]. A long line of research work around activity recognition reviewed in, e.g. [YWC08; HO20], has focused on the problem of optimal placement and combination of sensors on the body in order to improve *a priori* models' performance. Additionally, domain models derived from biomechanical studies like [MSB20; Car+18] are often used to describe body movements and the relative interactions between various body parts in a structured manner. Alternatively, considering the structure of the sensing devices explicitly during the learning process is more promising but challenging. An approach close to ours for the relativity of perspectives is that of [Est+19] which describes the different perspectives by discrete subgroup of the rotation group.

Integrating these additional models into the learning process has promising implications noticeably for the conciliation process of decentralized machine learning algorithms: one can exhibit the relative contribution of the individual views to the bigger picture. The primary goal of this chapter is to develop a robust approach that integrates knowledge about the structure of sensing devices in a principled way to achieve better collaboration.

6.2 Multi-level abstraction of the source position

Here, we propose an instantiation of the proposed problem formulation composed of local learners and a central model. To perform the separation of the position-specific components from the universal ones, we use a family of models based on variational autoencoders (VAEs) [KW13] (§ 6.2.1). The central model is responsible for aggregating the progress made by the local learners (§ 6.2.2). This instantiation is described in the following. Figure 6.2 summarizes the proposed instantiation. Algorithm 10 outlines the complete learning process.

6.2.1 Position-specific learners

The position-specific (or local) learners, denoted L_p, pursue their own learning steps locally using their own generated data (see the black arrows depicted in

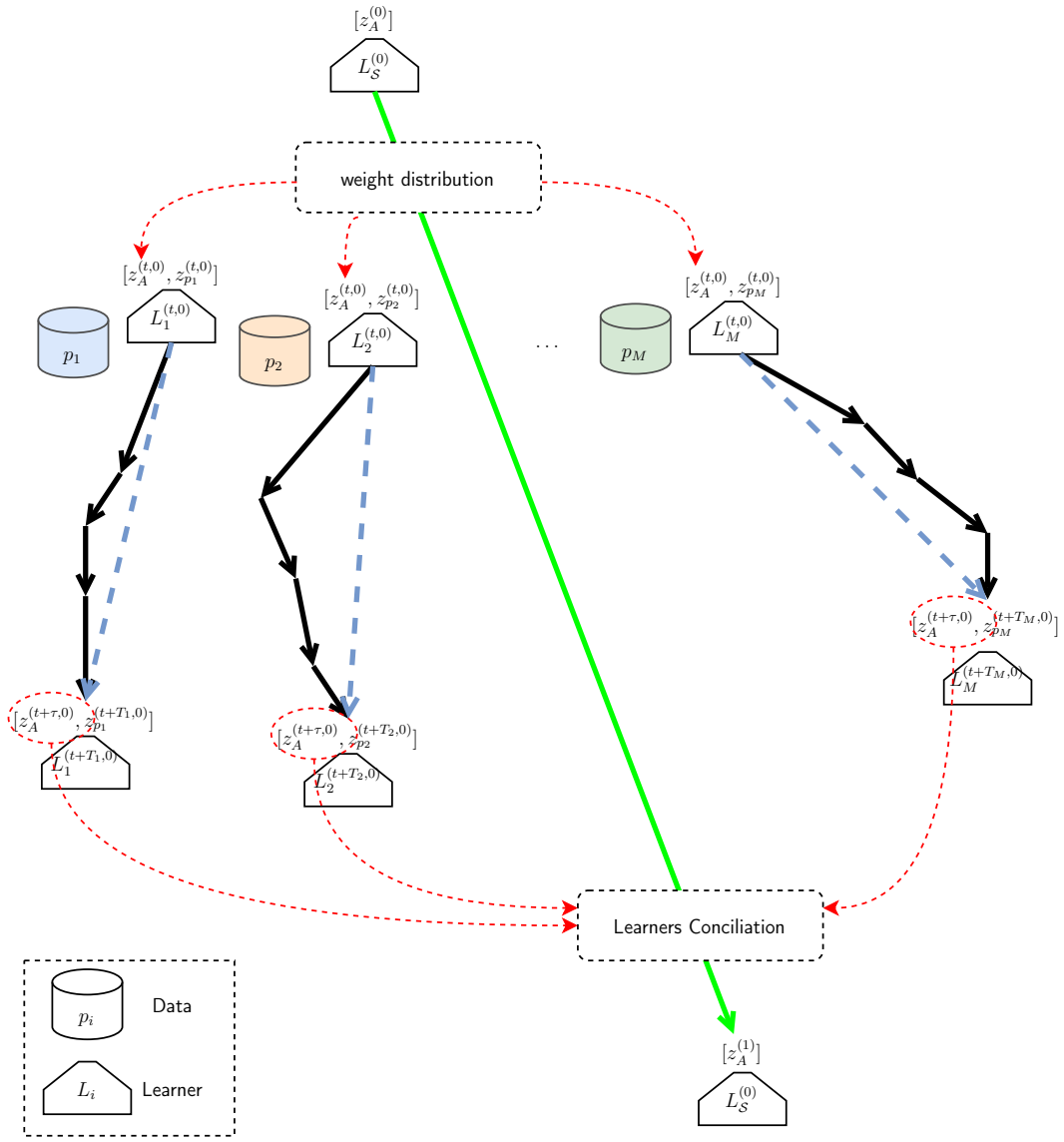


Figure 6.2: The framework of the proposed multi-level abstraction architecture. The global learner $L_{\mathcal{S}}$ (or L_{ref}) starts with an initial set of weights that are distributed to the local learners. The local learners L_p, one for each position p_i, learn the two vector components z_A and z_P by independently performing a set of gradient steps which allows getting newer versions. These new versions are used during the conciliation step, which results in a new version of the global learner and, subsequently, a more robust position-independent representation.

Figure 6.2). Their goal is to decompose the contents of the data into different factors of variations, particularly those related to the position itself.

The task here is to learn these factors of variation, commonly referred to as learning a disentangled representation (or transformation). It corresponds to finding a representation where each of its dimensions is sensitive to the variations of exactly one precise underlying factor and not the others. We construct at the level of each client i a representation that maps the observation space X to a latent space V with $h_A : X \to V$ (universal) and $h_{p_i} : X \to V$ (position-specific). The universal representation has to remain invariant to the relative location of the decentralized nodes. We also ensure during the learning process that the universal and location-specific transformations are orthogonal to each other ($h_A \perp h_{p_i}$). In other words, we want these two transformations to capture completely different factors of variations in the data. To do that, we enforce h_{p_i} to be insensitive to the factors of variations linked to the representation h_A using representation disentanglement techniques. The local objective function is constructed using a family of models based on VAEs for their ability to deal with entangled representations.

VAE-based objective. Depending on the availability of explicit knowledge about the underlying factors of variation, different strategies are pursued to learn the disentangled representation. For example, in video prediction [DB17; Hsi+18], temporal-invariance is often leveraged with a content representation that captures structure that is shared across all video frames and a pose representation capturing content that varies over time. These strategies require devising complex architectures and intricate loss functions to enforce prior knowledge. Alternatively, the disentanglement can be performed using separate representations for each factor of variation, which are jointly learned by different encoders, e.g. [Sad+20; Qia+21]. Although the representations are explicitly separated and learned by different encoders, getting exact correspondence with the factors of variation, i.e., non-overlapping dimensions, is not ensured and can lead to identical representations.

Recent advances in unsupervised disentangling based on VAEs demonstrated noticeable successes in many fields using the β-VAE, which leads to improved disentanglement [Hig+17]. It uses a unique representation vector and assigns an additional parameter ($\beta > 1$) to the VAE objective, precisely, on the Kullback Leibler (KL) divergence between the variational posterior and the prior, which is intended to put implicit independence pressure on the learned posterior. The improved objective becomes:

$$\begin{aligned}\ell_{p_i}(\mathbf{x}^i; \psi^i, \varphi^i) =& \mathbb{E}_{Q_{\varphi^i}(z^i|\mathbf{x}^i)}[\log P_{\psi^i}(\mathbf{x}^i|z^i)] \leftarrow \text{autoencoder reconstruction term} \\ &- \beta D_{KL}(Q_{\varphi^i}(z^i|\mathbf{x}^i)||P_{\psi^i}(z^i)) - \alpha D_{KL}(Q_{\varphi^i}(z^i)||P(z^i)),\end{aligned}$$

where the inference model $Q_{\varphi^i}(z^i|\mathbf{x}^i)$ corresponds to an encoder, and the likelihood model $P_{\psi^i}(\mathbf{x}^i|z^i)$ corresponds to a decoder. φ^i and ψ^i are the parameters of the encoder and decoder, respectively. The term controlled by α allows specifying a much richer class of properties and more complex constraints on the dimensions of the learned representation other than independence. Indeed, the proposed conciliation step is challenging due to the dissimilarity of the data distributions across the local learners, leading to discrepancies between their respective learned representations. One way to deal with this issue is by imposing sparsity on the latent representation in a way that only a few dimensions get activated depending on the learner and activities. We ensure the emergence of such sparse representations using the appropriate structure in the prior $P(z)$ such that the targeted underlying factors are captured by precise and homogeneous dimensions of the latent representation. We set the sparse prior as $P(z) = \prod_d (1-\gamma)\mathcal{N}(z_d; 0, 1) + \gamma\mathcal{N}(z_d; 0, \sigma_0^2)$ with $\sigma_0^2 = 0.05$ and d denotes the latent dimension. This distribution can be interpreted as a mixture of samples being either activated or not, whose proportion is controlled by the weight parameter γ [Mat+19].

6.2.2 Referential learner

Each local learner pursues its own version of the universal representation but has not to diverge from the universal representation z_A^{ref} aggregated at the level of the referential learner, denoted ref. The referential universal representation constitutes a consensus among all local learners. In our setting, we build the referential (or central) universal representation by making every learner contributes to it via a weighted aggregation defined as follows: given the objectives $f_i(\theta_i)$ of the local learners, the referential learner objective function is formulated as:

$$\min_{\theta \in \Theta} \left\{ f(\theta) := \frac{1}{M} \sum_{i=1}^{M} \alpha_i \times f_i(\theta_i) \right\} \text{ with } \sum_{i=1}^{M} \alpha_i = 1, \tag{6.1}$$

where α_i is used to weigh the contribution of every learner to the universal representation. After a predefined number of local update steps, we conduct a conciliation step (see the dotted arrows in Figure 6.2). Each conciliation step t produces a new version of the referential learner $\theta_{\mathsf{ref}}^{(t)}$ and, a new version of the referential universal representation z_A^{ref}. The conciliation step has to be performed on the learned representations z_A^p via regularization, for example. In our approach, the conciliation step is performed via representation alignment, e.g., correlation-based alignment [And+13]. More formally, we instrument the objective function of the local learners with an additional term derived from the representation align-

ment [TTN20]. The optimization problem (3.1) becomes:

$$\min_{\theta \in \Theta} \left\{ f(\theta) = \frac{1}{M} \sum_{i=1}^{M} \alpha_i \big(f_i(\theta_i) + \lambda R(\theta_i) \big) \right\}, \tag{6.2}$$

where R is a regularization term responsible for aligning the locally learned universal components with the ones learned by the referential learner and $\lambda \in [0, 1]$ is a regularization parameter that balances between the local objective and the regularization term. Precisely, in multi-view representation learning, representation alignment is defined as follows:

$$f(x^a; \theta_f) \leftrightarrow g(x^b; \theta_g)$$

where each view has a corresponding transformation (f or g) that takes the original space into a multi-view aligned space with certain constraints, and $\leftrightarrow$ denotes the alignment operator. In the case of correlation-based alignment, which relies on the canonical correlation analysis (CCA) [Hot92], this operator is concerned with finding a pair of linear transformations such that one component within each set of transformed variables is correlated with a single component in the other set. This makes the corresponding examples in the two views maximally correlated in the projected space,

$$\rho = \max_{f,g} corr\big(f(x^a), g(x^b)\big)$$

where $corr(\cdot)$ denotes the sample correlation function. Maximizing the correlations between the projections of the examples allows obtaining an embedding that compensates for the pairwise deficiencies of the different views. The regularization term in (6.2) is defined as follows:

$$R(\theta) = \max_{\theta} corr\big(z_A^{\mathsf{ref}}, h_A^{p_i}(x^i; \theta)\big), \tag{6.3}$$

the first part being fixed to be the referential universal representation. Position-specific and universal components will still be learned separately but locally. Then, the conciliation can be performed, where the weights of the local universal components are aggregated and used to update the referential learner.

6.3 Experiments

Here, we perform an empirical evaluation of the proposed approach, consisting of two major stages. In the first stage, we evaluate the quality of the data separation into position-specific and universal components which is performed by the local

Algorithm 9: Multi-level abstraction of sensor position

Input : $\{\mathbf{x}^i\}_{i=1}^{M}$ streams of annotated observations from the sensors

$\theta \leftarrow$ initWeights() ; % *Init. referential learner weights*

distributeWeights(θ, $\mathcal{S}$) ; % *Weights distribution*

while *not converged* **do**

; % *Local updates*

foreach *position* $p_i \in \mathcal{S}$ **do**

for $t \in T_i$ *steps* **do**

Sample mini-batch $\{x_j\}_{j=1}^{n_i}$ from the stream of data $\mathbf{x}^i$

Evaluate $\nabla_{\theta_i} \ell_{p_i}(\theta_i)$ with respect to the mini-batch

Compute adapted parameters: $\theta_i^{(t)} \leftarrow \theta_i^{(t-1)} - \eta \nabla_{\theta_i} \ell_{p_i}(\theta_i)$

end

end

; % *Central updates*

Update central model's weights $L_{\mathcal{S}}$ by aggregating the incoming weights from the local models $L_i, i \in \{1, \ldots, M\}$.

end

Result: $L_{\mathcal{S}}$ and $L_i, i \in \{1, \ldots, M\}$, the trained referential and local learners

learners and how each of these components contribute individually, with and without the conciliation process, to the recognition performances (§ 6.3.2); We, then, evaluate various inference configurations where the position-specific and universal components are combined together to improve the recognition performances. We also provide a comparative analysis against baselines (§ 6.3.3).

6.3.1 Experimental setup

Datasets description. We evaluate our proposed approach on three large-scale real-world wearable benchmark datasets featuring multi-location and heterogeneous sensors: SHL, HHAR, and Fusion datasets.

- The Heterogeneity Dataset for Human Activity Recognition (HHAR) [Sti+15] provides data from smartphones and smartwatches built-in sensors specifically devised to investigate sensor-, device- and workload-specific heterogeneities on HAR models. The dataset features 2 types of modalities, i.e., accelerometer and gyroscope, sampled according to the highest sampling rate of the respective devices. A total of 6 activities carried by 9 different users were recorded, including *Biking*, *Sitting*, *Standing*, *Walking*, *Stair Up* and *Stair down*.

- Fusion dataset [Sho+14] containing accelerometer, gyroscope, linear acceleration and magnetometer from smartphones placed on *right upper arm*, *right wrist*, *belt*, *right and left jean pocket*. Data collection from all 5 positions was performed in a synchronized manner at a sampling rate of 50Hz. This dataset considers 8 different activities, including *walking*, *running*, *sitting*, *standing*, *jogging*, *biking*, *walking upstairs* and *walking downstairs*, which were performed by 10 participants.

Baselines. Similar to the previous chapters, we compare our proposed approach with the following closely related baselines: DeepConvLSTM, DeepSense, and AttnSense (see § 4.3.1 for details). For the ablation study, we also compare our approach with two baselines that do not perform the separation nor conciliation steps. For each position, the architecture of these basic models consists of convolution-based circuits which are then fused together and trained jointly. We implemented two types of fusion schemes: concatenation-based and alignment-based fusion.

- **Concatenation-based alignment**: the outputs of the convolution-based circuits of each position are fused using a simple concatenation layer.
- **Correlation-based alignment**: the circuits' outputs are fused using a correlation-based conciliation layer [And+13] which allows the circuits to compensate for each other's deficiencies.

To make these baselines comparable with the models based on our proposed solution, we make sure to get the same complexity, i.e., comparable number of parameters.

Implementation details. For the closely related baselines, the available implementation is used otherwise, we reproduce them. We use Tensorflow [Aba+16] for building the architecture of the VAE used to model the learners in our proposed approach. This architecture is illustrated in Figure 6.3. As a preprocessing step, the annotated input streams are segmented into sequences, e.g., in the case of the SHL dataset, we obtain sequences of 6000 samples which correspond to a duration of 1 min. given a sampling rate of 100 Hz. To model the temporal dependencies in the considered sequences, we use LSTM cells [HS97]. For weight optimization, we use stochastic gradient descent with Nesterov momentum of 0.9 and a learning rate of 10^{-1}. Weight decay is set to 10^{-5}. The number of update steps τ_p performed by each local learner before the conciliation step is set to 100.

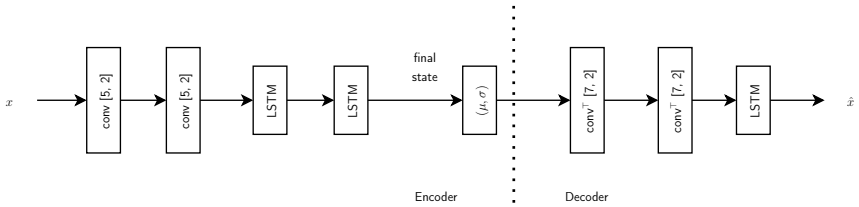


Figure 6.3: Example of VAE architectures used to model the learners. All convolutions are 1D with their hyperparameters (kernel size and stride) shown. All layers are preceded by batch normalization and a ReLU activation. conv$^\top$ stands for transposed convolution. LSTM cells are used to capture the temporal dependencies in the considered sequences. The final states generated by the LSTM cell are used to model the latent distribution's mean and variance. The number of layers and their hyperparameters are optimized.

6.3.2 Evaluation of the data decomposition process

In this part, we evaluate the ability of the local learners to decompose the sensor data into the position-specific components and the universal ones. We evaluate this process with and without the conciliation phase, then we show the impact of this step on the recognition performances. We measure the sparsity of a given representation using the *Hoyer* extrinsic metric [HR09] which is formally defined for a vector $\mathbf{y} \in \mathbb{R}^d$ to be:

$$Hoyer(\mathbf{y}) = \frac{\sqrt{d} - \|\mathbf{y}\|_1/\|\mathbf{y}\|_2}{\sqrt{d} - 1} \in [0, 1]$$

yielding 0 for a fully dense vector and 1 for a fully sparse vector. Table 6.1 summarizes the average normalized sparsity of the obtained representations. Figure 6.4 illustrates the average latent magnitude computed for each dimension of the learned representations.

Config.	*Bag*	*Hand*	*Hips*	*Torso*
w/o concil.	0.42±.072	0.77±.002	0.71±.029	0.68±.024
w/ concil.	0.44±.0145	0.91±.0521	0.87±.038	0.727±.033

Table 6.1: Summary of the per-position average normalized sparsity measured using the *Hoyer* extrinsic metric. Results with and without the conciliation step are shown.

From Table 6.1, we can observe, as expected, that the representations learned

by the local learners of the *Hand* and *Hips* have high sparsity compared to *Bag* and *Torso*. The sparsity increases further when the conciliation is performed as the dimensions that are less important are being pushed more and more towards zero. Regarding the latent magnitudes, during conciliation, we can observe that some dimensions of the latent representation of the central learner are getting more and more activated (e.g., dimensions 30, 35, 39, and 40 with an average magnitude of 0.0134, 0.146, 0.0138, and 0.138, resp.) corresponding to the universal components, while the remaining dimensions having low activation and some noticeable picks (e.g., at 3, 12, 18, and 24) corresponding to the position-specific components.

Evaluation of the recognition performances. As demonstrated above, the dimensions of the learned representations have a meaningful interpretation with regard to the activities that we seek to recognize. To further assess the usefulness of the separated components per se (without a conciliation step), we leverage them in a traditional discriminative setting. In other words, we take the learned representation and add, on top of it, a simple dense layer. This additional layer is trained to minimize classification loss while the rest of the circuit is kept frozen. Note that the additional Dense layer has a low VC dimension so that we ensure it has no capacity to improve the representation by itself. Table 6.5 compare the recognition performances obtained with the baseline models on the considered representative datasets. Furthermore, to better understand how the process of conciliation among the learners, attached to the different positions, impacts the quality of both the universal and position-specific components, we leverage similarly the separated components but this time, after performing the conciliation process. Table 6.3, summarizes obtained results. We compare the results with baseline models trained on data generated from specific positions without applying the separation nor conciliation processes. This configuration is referred to as *Baseline (no sep.)* in Table 6.3.

Model	HHAR	Fusion	SHL
DeepConvLSTM	70.1±.0018	68.5±.002	65.3±.0206
DeepSense	72.0±.0022	69.1±.0017	66.5±.006
AttnSense	76.2±.0074	70.3±.0027	68.4±.03
Feature fusion	72.9±.004	68.7±.001	66.8±.009
Corr. align.	75.8±.0014	70.2±.04	69.1±.015
Proposed	78.3±.0045	72.8±.002	74.5±.0133

Table 6.2: Recognition performances of the baseline models on different representative related datasets.

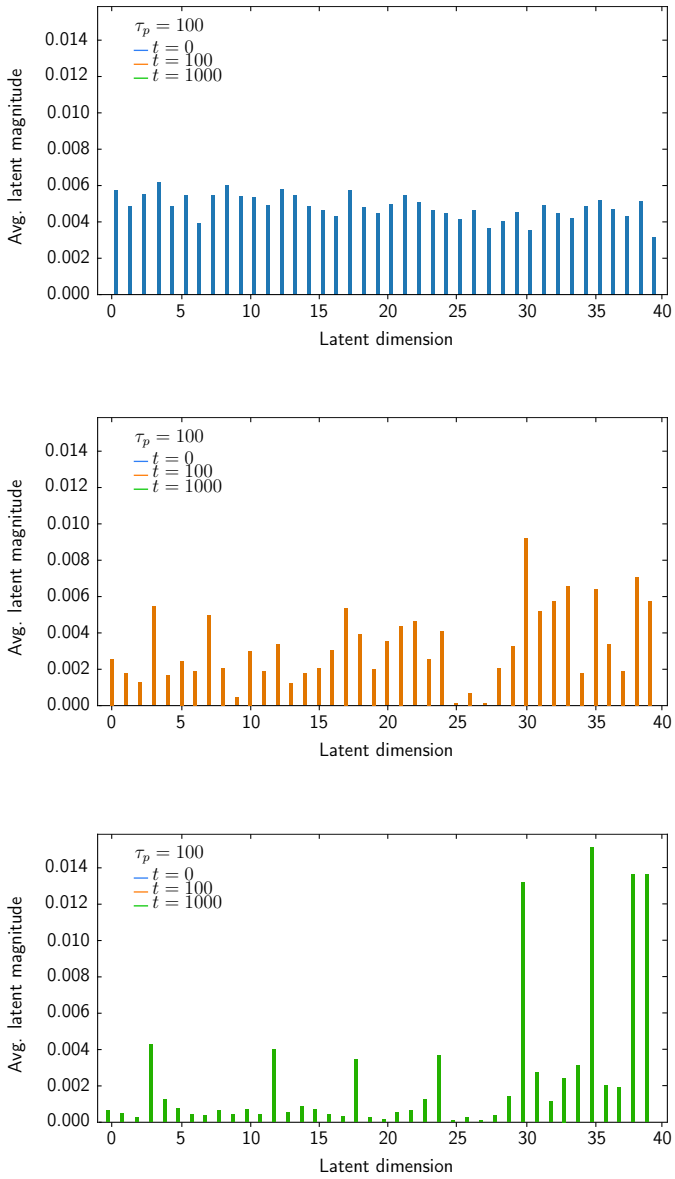


Figure 6.4: Average latent encoding magnitude computed over different steps of the conciliation process.

We observe from Table 6.3 that, overall, the recognition performances obtained using the position-specific and universal components are better than those obtained using the baseline (without separation nor conciliation). In theory, with the conciliation step, optimal representations would emerge in particular for the universal components. Indeed, this is achieved by the additional alignment term in Eq. 6.2

Config.	*Bag*	*Hand*	*Hips*	*Torso*
Baseline (no sep.)	63±.0089	63±.0014	65±.0126	60±.0072
Universal comp.				
w/o conciliation	66±.0224	65±.0147	66±.0035	62±.013
w/ conciliation	66±.016	67±.0015	67±.0354	63±.01
Pos.-specific comp.				
w/o conciliation	64±.3	66±.007	67±.0026	61±.087
w/ conciliation	65±.029	68±.03	70±.07	61±.029

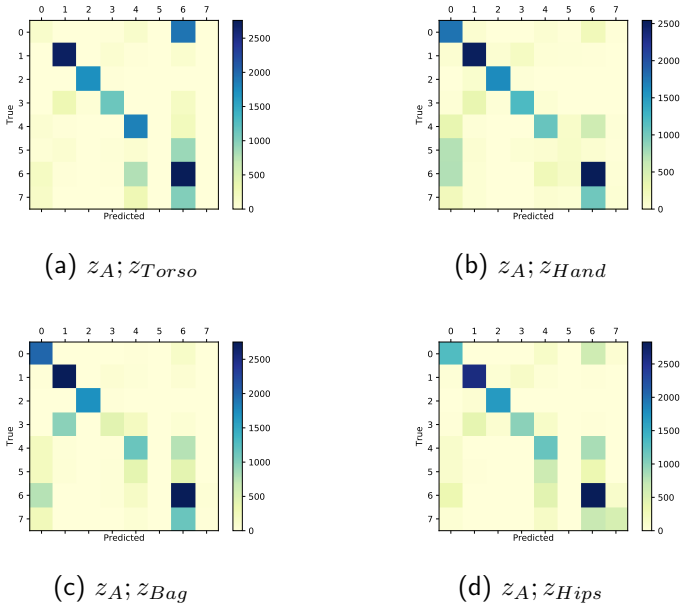
Table 6.3: Summary of the recognition performances obtained using either the universal or the position-specific components learned in each position by the local learners. Recognition performances with and without the conciliation process are reported. For reference, the recognition of a baseline model which do not perform separation (nor conciliation) are additionally shown.

which should make them interchangeable regardless of the position from which they have been generated. This should nevertheless be harder in the case of the position-specific components which may activate very diverse dimensions of the learned representation (as described in the experimental results above). Surprisingly, this has a mild impact on the recognition performances which stay comparable. This could potentially be explained by the importance of the position-specific components for the recognition of many of the activities that are considered in the SHL dataset. It is worth noticing though that the universal components achieve remarkable improvements in the case of *Bag* and *Torso*.

6.3.3 Inference configurations

Here we evaluate the robustness of the proposed approach to the evolution of the sensors deployments via the flexibility that it offers for the inference step. Depending on the activity, the right prediction can be achieved by using either components z_A or z_P taken individually or a combination of the universal component z_A and the most appropriate position-specific component. In this part, we take a fine-grained look at the previously obtained recognition performances by assessing the optimal configuration, which allows the correct prediction of each of the individual activities we are interested in. For this, we evaluate the predictions obtained using basic inference configurations, i.e., the combination of the universal components with *Torso*-specific components $[z_A; z_{Torso}]$; *Hand*-specific components $[z_A; z_{Hand}]$; *Bag*-specific components $[z_A; z_{Bag}]$; and with *Hips*-specific components $[z_A; z_{Hips}]$.

Figure 6.5 shows the confusion matrices obtained using each of these configurations.



(a) $z_A; z_{Torso}$ (b) $z_A; z_{Hand}$

(c) $z_A; z_{Bag}$ (d) $z_A; z_{Hips}$

Figure 6.5: Confusion matrices obtained using different inference configurations. Combination of the universal components z_A and: (a) *Torso*-specific components; (b) *Hand*-specific components; (c) *Bag*-specific components; (d) *Hips*-specific components. The activities are numbered as *1:Still*—*8:Subway*.

Compared to the baseline models, the evaluated inference configurations yield better recognition performances in general. For example, the combination of the universal and most of the position-specific components help discriminate activities like *Walk*, *Run*, and *Bike* efficiently. On the other hand, some activities like *Car*, *Bus*, or *Train* suffer from confusion and do no show significant improvements over the baseline (approx. 2% on avg.). Also, activity *Subway* exhibits the same behavior with less proportion suggesting that this "on-wheels" group of activities needs an elaborate combination of points of view. This issue could potentially be circumvented by using more featured inference configurations where other position-specific representations (or learners), rather than a single one, can be leveraged to infer these problematic or hard-to-infer activities.

Table 6.4 summarizes the evaluation results of the inference configurations featuring the combination of various position-specific components. We observe an increase in terms of the correct predictions for most of the activities compared to the previous setting. In particular, the "on-wheels" group of activities, i.e.,

Activities	*Best Config.*	*Perf.±std.*	*mean±std.*
Still	$z_{hi}; z_t$	85.77±0.016	83.26±0.7
Walk	$z_A; z_{ha}$	88.54±0.07	86.74±0.058
Run	z_{ha}	90.51±0.016	89.46±0.03
Bike	$z_A; z_{hi}$	85.62±0.2	83.22±0.086
Car	$z_A; z_{ha}$	78.24±0.058	77.14±0.2
Bus	z_{ha}	78.08±0.022	75.17±0.004
Train	$z_{hi}; z_{hi}$	76.13±0.175	74.88±0.08
Subway	$z_A; z_{ha}; z_t$	75.89±0.009	74.07±0.006

Table 6.4: Recognition performances (mean and std.) of the best configurations is shown, along with the recognition performances (mean and std.) averaged over all evaluated ones (repeated 7 times). The ubscripts of the position-specific representations are shortened as z_b (*Bag*), z_{ha} (*Hand*), z_{hi} (*Hips*), and z_t (*Torso*).

car, *bus*, *train*, and *subway*, get improved substantially. At the same time, as expected, we see now that the inference configurations, which yield the highest recognition performances for these activities, use genuine combinations like *Hand*-specific components alone in the case of *Bus* or a combination of the universal, *Hand*- and *Torso*-specific components in the case of *Subway*. On the other hand, *Still* gets the least improvement compared to the previous setting, while the best configuration to infer it is a combination of the *Hand*- and *Torso*-specific components (85.77±0.016). It is worth noticing that activities like *Walk* and *Bike* still achieve competitive performances (88.54±0.07 and 85.62±0.2, resp.) while using the same inference configuration, i.e., a combination of the universal and *Hand*-specific components for *Walk* and *Hips*-specific for *Bike*, as in the previous setting. In the case of *Run*, the highest recognition performances are achieved using only the *Hand*-specific components, which supports the observations presented in the introduction to this chapter.

6.4 FEDABSTRACT algorithm

In this part, we present a novel approach that abstracts the exact context surrounding the data generators. Local learners are trained to decompose, as before, the learned representations into (i) universal components shared across devices and locations and (ii) local components that capture the specific device- and location-dependent context. Besides this decomposition process, we leverage knowledge about the structure of the sensing deployment by representing the relative geom-

etry of the sensing devices with group transformations. Indeed, we introduce the notion of relativity between data generators and model it via the special Euclidean group, denoted by $SE(3)$. It encompasses arbitrary combinations of translations and rotations, which are used to express the relative contribution of a data generator to describing (or learning) a phenomenon of interest. This way, the learner is constrained using principled mathematical tools, and the symmetry structure induced by the relative data generators is reflected effectively in the latent space.

In the following, we describe this approach in detail. At a given decentralized location, there are three different elements that are learned: (1) the universal (or group-invariant) and (2) position-specific representations (§6.4.1), and (3) the group of relative geometry representation (§6.4.2), The generalization capabilities of the universal representation are improved collaboratively across the decentralized sensing devices via the conciliation (or aggregation) process (§**??**). Figure 6.6 summarizes the proposed approach. Extensive experiments on two large-scale real-world wearable benchmark datasets featuring structured sensing environments are presented to assess the effectiveness of the proposed approach.

6.4.1 Learning group-invariant and position-specific representations

The idea is to express the data generated from a decentralized device (e.g., hand sensors in the case of on-body sensor deployments) relative to the coordinate system of a referential (e.g., torso.) This way, the exact relative contribution of the sensing device is captured without the contextual artifacts. To do this, we have to capture the variations due to the relative location of the decentralized device w.r.t. a global coordinate system and capture invariant aspects that are shared across the devices. The latter aspects are universal components that are shared with the central model, while the former ones are considered as specific components which add noise to the learning process, thus requiring to be discarded from it.

Learning h_A and h_{p_i} locally

The data x_i captured at a given location p_i are generated from two underlying factors: one reflecting the position-specific components and the other the position-invariant (or universal) components. The task here is to learn these factors of variation, commonly referred to as learning a disentangled representation. In other words, we want these two transformations to capture completely different factors of variations in the data. To do that, we enforce h_{p_i} to be insensitive to the factors of variations linked to the representation h_A using representation disentanglement techniques. It corresponds to finding a representation where each of its dimensions is sensitive to the variations of exactly one precise underlying factor and not the

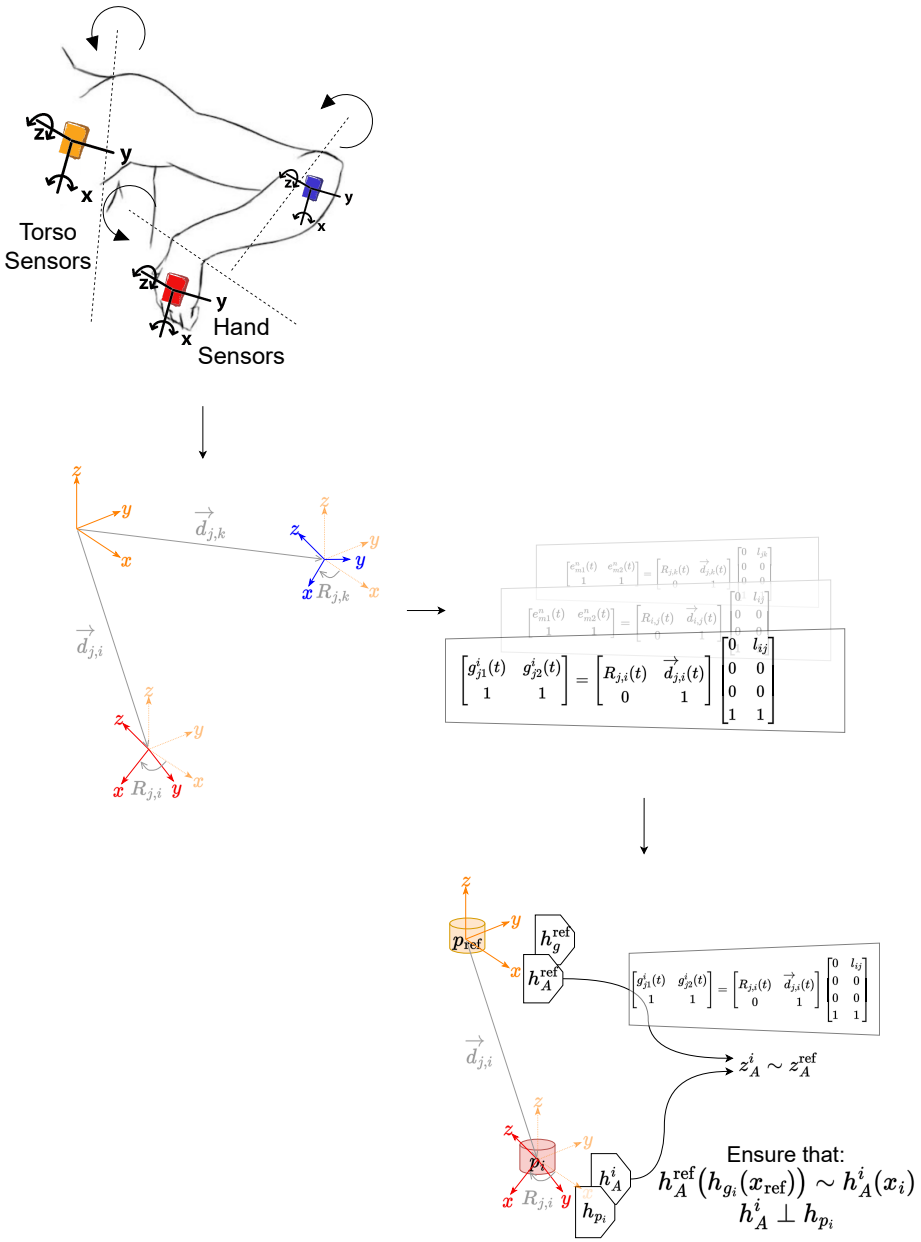


Figure 6.6: The framework of the proposed approach. Explicitly representing the relative geometry of the decentralized devices and their symmetries using elements of the special Euclidean group $SE(3)$ and leveraging them to constrain the learning process with the goal of reducing the problem size and improving data efficiency.

others. Note that the inputs to h_A in the local learners are the raw sensory data x_i generated locally.

At this point, we are again left with two alternatives for jointly learning the universal transformation h_A and the position-specific transformation h_{p_i} at the local learner level: (1) using a separate VAE for each transformation and training each one of them jointly using the raw sensory data as inputs; (2) using a single VAE and train it to automatically factorize the learned representation so that each axis captures specific components. We use the same objective as (6.1).

Now, we have to represent the concept of data generator relativity and its induced symmetries in the form of group elements whose action on the data leaves the universal component of the learned representation invariant.

6.4.2 Relative geometry for data generators

We model the relative geometry of sensors and the perspectives they provide via the special Euclidean group $SE(3)$. Let $\mathbf{x}^i$ and $\mathbf{x}^j$ be the stream of observations generated by the data sources s_i and s_j. At each time step t, the observations x_i and x_j generated by these data sources are related together via an element $g_j^i \in SE(3)$ of the group of symmetries, i.e., the observation x_i is obtained by applying g_j^i on x_j. Here, we want to learn a mapping h_{g_i} for each decentralized device so that the biases that stem from the context (exact position) are corrected before its contribution is communicated to the global model.

Special Euclidean group $SE(3)$. The special Euclidean group, denoted by $SE(3)$, encompasses arbitrary combinations of translations and rotations. The elements of this group are called rigid motions or Euclidean motions and correspond to the set of all 4 by 4 matrices of the form $P(R, \overrightarrow{d}) = \left(\begin{smallmatrix} R & \overrightarrow{d} \\ 0 & 1 \end{smallmatrix}\right)$, with $\overrightarrow{d} \in \mathbb{R}^3$ a translation vector, and $R \in \mathbb{R}^{3\times 3}$ a rotation matrix. Members of $SE(3)$ act on points $z \in \mathbb{R}^3$ by rotating and translating them: $\left(\begin{smallmatrix} R & \overrightarrow{d} \\ 0 & 1 \end{smallmatrix}\right)\left(\begin{smallmatrix} z \\ 1 \end{smallmatrix}\right) = \left(\begin{smallmatrix} Rz + \overrightarrow{d} \\ 1 \end{smallmatrix}\right)$.

Relative geometry representation. Given a pair of sensing devices s_i and s_j located at positions p_i and p_j, each having its own local coordinate system attached to it. We represent the relative geometry of this pair by expressing each of the devices in the local coordinate system of the other (see Figure 6.7). Similarly to [VAC14], the local coordinate system attached to p_i is the result of a translation $\overrightarrow{d}_{j,i}$ and a rotation $R_{j,i}$, where the subscript j, i denotes the sense of the transformation being from p_j to p_i. While the translation corresponds to the alignment of the origins of the two coordinate systems, the rotation is obtained by rotating the global coordinate system such that the x-axis of the two coordinate

systems coincide:

$$\begin{pmatrix} g^i_{j1}(t) & g^i_{j2}(t) \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} R_{j,i}(t) & \overrightarrow{d}_{j,i}(t) \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & l_{ij} \\ 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{pmatrix}. \tag{6.4}$$

The relative geometry of the data generators is considered to be elements of $SE(3)$ and supposed to capture the transformations acting on the data generators. Without explicit information about the exact locations of the data generators, these transformations have to be learned. For this, we parameterize the transformation matrices used to represent the relative geometry of the data generators with learnable weights. In particular, we parameterize as in [QBC20] the n-dimensional representation of a rotation R as the product of $\frac{n(n-1)}{2}$ rotations, denoted $R^{v,w}$, each of which corresponds to the rotation in the v, w plane embedded in the n-dimensional representation. For example, a 3-dimensional representation has three learnable parameters, $g = g(\theta^{1,2}, \theta^{1,3}, \theta^{2,3})$, each parameterizing a single rotation, such as $R^{1,3}(\theta^{1,3}) = \begin{pmatrix} \cos\theta^{1,3} & 0 & \sin\theta^{1,3} \\ 0 & 1 & 0 \\ -\sin\theta^{1,3} & 0 & \cos\theta^{1,3} \end{pmatrix}$.

Learning h_A and h_g in the central server

The referential learner (or central server) happens also to be a learner similar to the local learners. The main difference is that the referential learner is located in a particular position of the sensors deployment, i.e., the referential coordinate system, which imposes it to perform additional processing. Let's denote the referential learner with subscript ref (the orange data source in Figure 6.7). The referential learner maintains the specific h_g's corresponding to each individual position of the sensors deployment and ensures that:

$$h_A(h_{g_i}(x_{\mathsf{ref}})) = h_A(x_i), \forall i \tag{6.5}$$

where h_{g_i} is the learned representation corresponding to the group action acting on the data x_i generated by the sensor located at position i and x_{ref} is the data generated by the sensor located at the referential point. The h_{g_i} transformation is learned by the referential learner using the raw data generated at the central server level. The constraint imposing the invariance, i.e., $h_A(h_{g_i}(x_{\mathsf{ref}})) = h_A(x_i), \forall i$, is the pivotal element that makes it possible to effectively learn this transformation.

By drawing a parallel with the construction of manifolds in latent spaces, this transformation can be interpreted as an operator projecting the data, generated by the data source positioned on ref, towards a latent space shifted by the action of the group elements so that the universal components learned by the transformation h_A (at the referential) coincide with those transformations $(h^i_A, \forall i)$ learned by the

local learners attached to the other positions. h_g must therefore act on different subgroups of the latent space. We ensure that the learned universal transformation h_A is invariant to the action of the group $SE(3)$, i.e., $h_A(gx) = h_A(x), g \in SE(3)$. For this we map the group $SE(3)$ to a linear representation GL on V, i.e., $\rho : SE(3) \to GL(V)$. Our goal is to map observations to a vector space V and interactions to elements of $GL(V)$ to obtain a disentangled representation of the relative geometry.

As there are many different group representations (one for each position of the deployment of the sensors) at the referential learner's level, we have to ensure that the learned representation h_g acts on specific subspaces of the latent space. At the central server, each client is considered to generate a subgroup of relative geometry. During the learning process, each subgroup of the symmetry group is made to act on a specific subspace of the latent space. Formally, let $\cdot : G \times X \to X$ be a group action such that the group G decomposes as a direct product $G = G_1 \times G_2$. According to [Hig+18], the action is disentangled (w.r.t. the decomposition of G) if there is a decomposition $X = X_1 \times X_2$, and actions $\cdot_i : G_i \times X_i \to X_i$, $i \in \{1, 2\}$ such that: $(g_1, g_2) \cdot (v_1, v_2) = (g_1 \cdot_1 v_1, g_2 \cdot_2 v_2)$, where $\cdot$ denotes the action of the full group, and the actions of each subgroup as $\cdot_i$. An G_1 element is said to act on X_1 but leaves X_2 fixed, and vice versa. We end up here in the same situation as in the disentanglement of universal and position-specific components, i.e., either we use a separate VAE for each group transformation or a single one for all the groups with the additional constraint stating that the action of each subgroup act on specific regions of the latent space manifold and leave the other regions fixed. This can be achieved via clustering of the latent space using a Gaussian mixture prior [Mat+19] $P(z) = \sum_{c=1}^{C} \pi^c \prod_d \mathcal{N}(z_d|\mu_d^c, \sigma_d^c)$, with C the number of desired clusters and π^c the prior probability of the c-th Gaussian.

At the local learner's level, the proposed model is trained in an end-to-end fashion. The generalization capabilities of the representation h_A are improved via the conciliation process performed across the nodes of the deployment. Algorithm 10 summarizes the process of the proposed approach and Figure 6.7 illustrates its bigger picture.

6.5 Experiments

We perform an empirical evaluation of the proposed approach, consisting of two major stages: (1) we verify the effectiveness of the proposed approach in the HAR task via a comparative analysis which includes representative related baselines (§6.5.1); (2) we also conduct extensive experiments and ablation analysis to demonstrate the effectiveness of the various components of our proposed approach (§6.5.2).

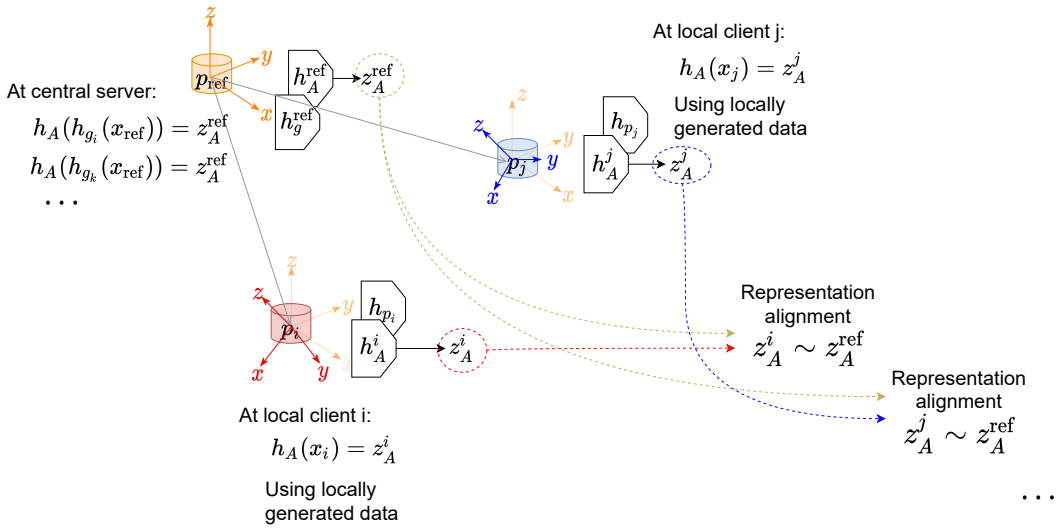


Figure 6.7: Network architecture of FedAbstract. The local learners (red and blue) perform a set of updates on their proper version of the universal representation. The referential learner at position p_{ref} (in orange) maintains the specific h_g's corresponding to each individual position of the sensors deployment and ensures that: $h_A(h_{g_i}(x_{\text{ref}})) = h_A(x_i), \forall i$ where h_{g_i} is the learned representation corresponding to the group elements acting on the data x_i generated at position i and x_{ref} the data generated at the referential point. Notice that only gradient updates are shared to the central server and the data generated at a given location is processed exclusively by the local learner.

Experimental setup. We evaluate our proposed approach on two large-scale real-world wearable benchmark datasets featuring structured sensing environments: SHL [Gjo+18] and Fusion [Sho+14] datasets. Here, we focus our evaluations on the SHL and Fusion datasets, which feature geometrical aspects related to the location of the data sources. Besides the closely related baselines used in the previous experimental part (§ 6.3), we compare our current approach against GILE [Qia+21]. GILE proposes to explicitly disentangle domain (or position)-specific and domain-agnostic features using two encoders. To constrain the disentanglement process, their proposed additional classifier is trained in a supervised manner with labels corresponding to the actual domain to which the learning examples belong. By analogy to GILE, in our approach, the domain labels correspond to the exact location of the data sources. In addition, we use our previous approach as a baseline. We refer to it as *FedAbstract, no* $SE(3)$ as it does not consider the relative geometry of the data generators.

To make these baselines comparable with our models, we make sure to get the

Algorithm 10: FEDABSTRACT Algorithm

Input : $\{\mathbf{x}^i\}_{i=1}^M$ streams of annotated observations

$\theta \leftarrow$ initWeights() ; % *Initialize global learner's weights*

distributeWeights(θ, $\mathcal{S}$) ; % *Weights distribution*

while *not converged* **do**

  foreach *position* p_i **do**

    for $t \in T_i$ *steps* **do**

      Sample mini-batch $\{x_j\}_{j=1}^{n_i}$

      Evaluate $\nabla_{\theta_i} \ell_{p_i}(\theta_i)$ w.r.t. the mini-batch

        └ Subject to $R(z_A^i, z_A^{\text{ref}})$ (e.g., correlation-based alignment [And+13])

      $\theta_i^{(t)} \leftarrow \theta_i^{(t-1)} - \eta \nabla_{\theta_i} \ell_{p_i}(\theta_i)$

      Ensure $h_A \perp h_{p_i}$ (see §6.4.1)

    end

    Communicate θ_A^i (with $\theta_i = [\theta_A, \theta_{p_i}]$)

  end

  $\theta_A \leftarrow \theta_A + \sum_{i=1}^M \alpha_i \cdot \Delta\theta_A^i$; % *Central updates*

  Enforce group action disentaglement (6.5)

end

Result: Globally shared universal representation h_A

same complexity, i.e., a comparable number of parameters. We use the f1-score in order to assess the performances of the architectures. We compute this metric following the method recommended in [FS10] to alleviate bias that could stem from unbalanced class distribution. In addition, to alleviate the performance overestimation problem, we rely in our experiments on the meta-segmented partitioning proposed in [HP15] (see § 5.3.1).

6.5.1 Performance comparison

We conduct extensive experiments to evaluate the performance of the proposed algorithm in the following two settings: activity recognition (or classification) task and representation disentanglement. For the activity recognition setting, Table 6.5 summarizes the performance comparison of the baselines in terms of the f1-score obtained on the SHL and Fusion datasets. Here we assess the usefulness of the separated components per se by leveraging them in a traditional discriminative setting. In other words, we take the learned representation and add a simple dense layer on top of it. This additional layer is trained to minimize classification loss while the rest of the circuit is kept frozen. Experimental results show that

Model	Fusion	SHL (Acc.)	SHL
DeepConvLSTM	68.5 ±.002	64.4 ±.0078	65.3 ±.0206
DeepSense	69.1 ±.0017	64.8 ±.0033	66.5 ±.006
AttnSense	70.3 ±.0027	69.6 ±.0072	68.4 ±.03
GILE	71.7 ±.014	71.1 ±.035	69.0 ±.001
FedAbstract	75.7 ±.047	75.7 ±.047	77.3 ±.017

Table 6.5: Recognition performances (f1-score) of the baseline models on different representative related datasets. Evaluation based on the meta-segmented cross-validation.

the proposed approach exhibits superior performance compared to the baselines. The proposed method achieves promising improvements in terms of f1-score over the baseline methods. In particular, our proposed approach improves recognition performances by approximately 7-9% on Fusion and SHL, while the improvement of attention-based methods is only about 1-2%. Compared to GILE, our approach shows consistent improvement on the considered configurations. This demonstrates that leveraging knowledge about the structure of the deployment, instead of simply using domain labels corresponding to the exact location of the data sources, improves disentanglement and ultimately activity recognition.

In the representation disentanglement setting, we assess the separation quality between the universal and position-specific components as well as those related to the actions of each subgroup. For this, the average latent magnitude computed for each dimension of the learned representations constitutes an appropriate measure. Figure 6.8 illustrates the average latent magnitude computed for the group of relative geometry representation. It shows the activated latent dimensions depending on the subgroup of transformations (among Bag, Hand, and Hips) acting on the data sources. We can see in particular that specific dimensions are activated depending on the subgroup of transformations that are used to stimulate the learned representation. These dimensions are also independent of each other. Furthermore, in complementary experiments, one can observe the evolution of the dimensions of the central learner's latent representation where some of them are getting more activated than others, which is a sign of the emergence of the desired universal components shared across the learners.

6.5.2 Ablation study

To demonstrate the generalization and effectiveness of each component of our proposed approach, we further design and perform ablation experiments on the SHL

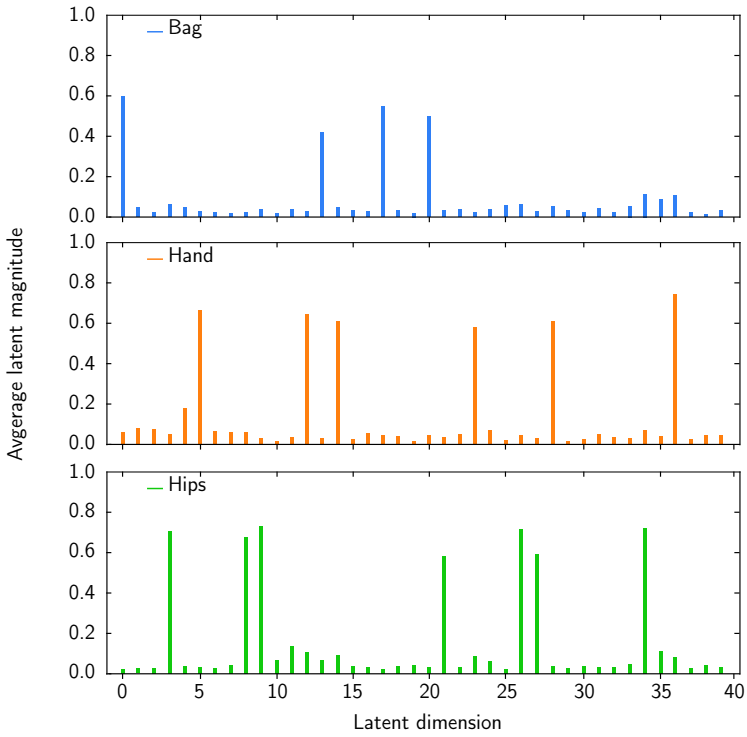


Figure 6.8: Average latent encoding magnitude in the SHL dataset. It shows the repartition of the latent dimensions being activated between the different subgroups of transformations acting on the data sources (Bag, Hand, and Hips positions).

and Fusion datasets. We compare FedAbstract to FedAvg [McM+17] and advanced solutions which try to correct for client-drift including SCAFFOLD [Kar+20]. FedAvg and SCAFFOLD do not perform explicit separation of the local data and thus constitute suitable baselines to assess the impact of each of FedAbstract's components. The experimental results illustrated in Figure 6.9 (top) are obtained using FedAbstract with both the relativity and decomposition constraints. These results suggest that the evolution of the loss in the case of FedAvg gets slower as we increase the number of local steps, which corresponds to the common observation that client drift increases proportionally to the number of local steps, hindering progress. At the same time, we observe that FedAbstract has excellent performance, slightly better than SCAFFOLD, suggesting a close connection between the estimate of the client-drift $\mathbf{c}_i$ and the position-specific components obtained via our proposed separation process.

Furthermore, we evaluate the effectiveness of explicitly representing the data generators' relativity via group actions while learning the universal and position-specific transformations. For this, we evaluate the performance of our proposed

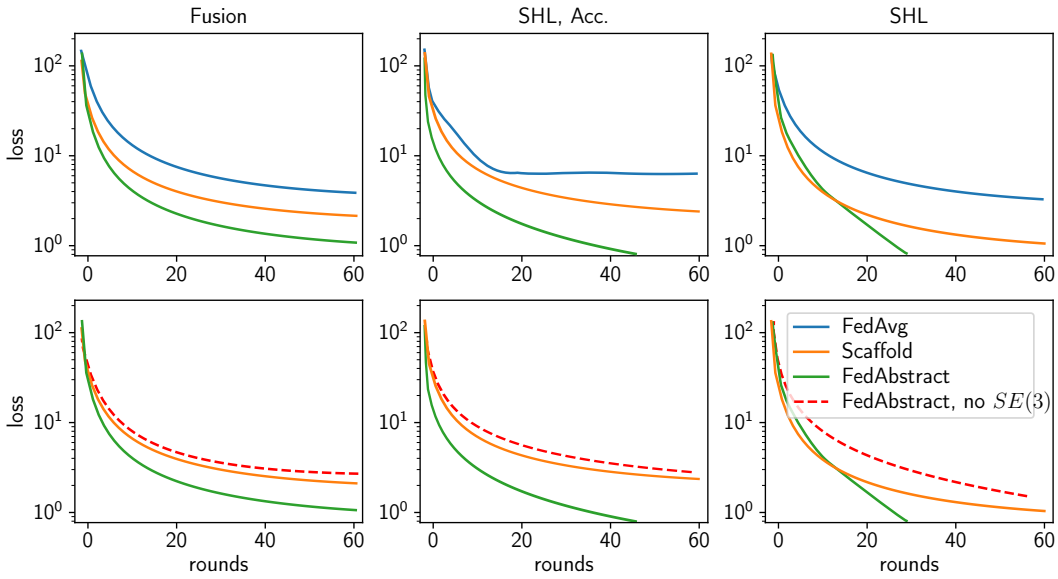


Figure 6.9: Evolution of the loss during decentralized learning. (top) FedAbstract with both the relativity and decomposition constraints. (bottom) FedAbstract without the relativity representation constraints (FedAbstract, no $SE(3)$).

approach against a setting that does not specifically consider the relative geometry of the data generators (FedAbstract, no $SE(3)$). Basically, in this setting, the constraint imposing the relative geometry is not enforced during the learning process. Figure 6.9 (bottom) illustrates the obtained results in terms of the loss evolution on both SHL and Fusion datasets. We notice that compared to the basic setting, enforcement of the relative geometry consistently improves the convergence by 5% on SHL and 3% on Fusion. We see that these differences correspond to the gap between SCAFFOLD and our proposed approach. This demonstrates that the separation process constrained by the explicit representation of relativity ultimately leads to improving collaboration across the decentralized devices.

6.6 Conclusion

We studied in the chapter the emergence of universal and context-specific components in the data generated in structured sensing environments. Sensors distributed in various positions of the space provide rich perspectives that need to be leveraged properly during the learning process. The information conveyed by these perspectives is not of the same nature, e.g., the sensor's position bias induces information of different types. Context-specific components correspond to

the context surrounding the data generators. For example, the relative location of a given data source in the sensor deployment or its corresponding sensing model (or characteristics) induces a context that has to be handled explicitly in order for the learning process to be effective.

At first, we proposed a multi-level processing framework, where local learners perform a disentanglement-based operation to separate the data into its constituting components, and a conciliation process at the central server allows for reinforcing the universal components via the aggregation of the locally learned universal components. The broader idea behind this framework is that *universal* components of the data are not directly accessible. However, it can be attained through various decentralized points of view. Collaboration is, therefore, not a confrontation but rather the accumulation of relevant symmetries and complementary information from each viewpoint whose contribution can be determined precisely. The model we propose achieves this. Indeed, experimental results show that the proposed approach substantially improves recognition rates and has many advantages, including the reduction of the heterogeneity impact, which is induced by the particular context within which the data sources are embodied. The data decomposition process allows a better recognition rate in several ways: (i) by reducing the noise induced by the data linked to the position itself, e.g., the local component of the movement of the hand constitutes noise for the local component of the movement of the feet; (ii) by aggregating only data of the same nature presenting different points of view and; (iii) for certain activities, the local component alone is sufficient to ensure recognition, e.g., the movement of a hand during the activity Running.

After that, building upon the multi-level framework for separating the local and universal components of the data, we proposed to further guide this process by explicitly enforcing a priori knowledge about the relativity of the data generators. We leverage for this additional knowledge in terms of symmetries and invariants that appear in these kinds of environments. These symmetries and invariants are explicitly represented in the form of group actions and incorporated into the learning process. In particular, we introduced the notion of relativity between data generators, and we modeled it via the special Euclidean group, denoted by $SE(3)$, which encompasses arbitrary combinations of translations and rotations. The relative contribution of a data generator in the description of the phenomena of interest is expressed using elements of this group and used to constrain the separation process. In particular, building on symmetry-based disentanglement learning, the symmetry structure induced by the relativity of the data generators is reflected in the learned latent space. This allows us to further leverage the notion of sharing, which is reflected in the conciliation process of the decentralized learning setting by promising improvements. Further, the proposed separation process

of the data into universal and position-specific components improves collaboration across the decentralized devices materialized by the conciliation (or aggregation) process. Obtained results on activity recognition, an example of real-world structured sensing applications, are encouraging and open-up perspectives for studying more symmetries, invariants, and also equivariants that emerge in these environments.

A direct extension of the work presented in this chapter includes leveraging these symmetries and invariants from a theoretical perspective like Lie group and corresponding algebra, a special and large class of continuous groups that includes many valuable transformations like translations, rotations, and scaling, and which also proposes a principled way for handling operations on the transformations such as composition, inversion, differentiation, and interpolation. Future work follows two axes: (1) improving the quality of the model, in particular, having a fine-grained control of the data decomposition process by adding additional domain knowledge-based models, e.g., representing explicitly the dynamics of the body movements in the latent space as done in [Kar+16; Wat+15] in the case of human activity recognition case; (2) the conducted research raises interesting questions to pursue, noticeably the improvement of multi-sources approaches where the various perspectives are entangled with local components, which could improve federated approaches by reducing the noise, especially by sharing only the mutualisable components.

We investigated in this chapter the collaborative aspects of the massively distributed sensing nodes (or data generators) by abstracting the biases induced by their surrounding context and integrating models of their relativity into the conciliation phase. The next chapter concludes this thesis.

Chapter 7

Conclusion

We conclude this study by summarizing the key research findings in relation to the research aims and questions we set in the introduction. We also discuss possible direct extensions to the proposed approaches, open problems, along with directions for future research.

Thesis summary

We addressed in this research the problem of learning in the context of the generalization and widespreadness of sensing, actuation, and computing capabilities, materialized, for example, by Internet of Things and Industry 4.0 applications. This context brings both practical and theoretical challenges that we proposed to address via meta-learning and modeling of the domain constraints.

We set up in the introductory chapter a long list of research challenges that we aimed to answer in this research. To name a few, we provided answers to the research questions regarding how to take into account domain-specific requirements and constraints in the learning process and how quantities of data needed to learn can be reduced so as to account for the transmission constraints and the cost of generating data. We also investigated ways to answer urging challenges related to the robustness of the learned models towards dynamical factors of the real-world deployments: for example, how to accelerate adaptation to these dynamical factors, and what do they induce in terms of deployment and monitored phenomena evolution? Similarly, aspects related to the transparency of the learning models in the context of generalized sensing, actuation, and computing capabilities have been discussed.

Let's reflect on the research we conducted in this thesis following different aspects: (i) collaboration in the massively distributed and decentralized context; (ii) domain knowledge that is leveraged and the way it is represented; (iii) modularity,

interpretability, and transparency of the learned models; and (iv) structural risk minimization, the introduction of structure in the set of admissible functions, and optimization landscape.

Collaboration in the massively distributed and decentralized context. Collaboration across the data sources (or, more generally, tasks) encompassing the distributed sensing environments is a natural process in this context and has been investigated, in this thesis, from different viewpoints. In Chapter 4, collaborative mechanisms during the learning process were implemented at the level of the learning examples used to learn. Indeed, the example selection and augmentation approaches proposed in this chapter are guided by domain knowledge and were particularly motivated to cope with variations across data sources in the sensing environments. We have seen that the variations across the data sources can be related to their disposition in space, their data-generating processes, and their sensing models. Knowledge about the way these data sources are structurally related to each other was ultimately leveraged to control how they collaborate with each other to learn a unified theory. In Chapter 5, collaboration materialized through the principles of transfer, sharing, and reuse across the levels of the hierarchies of concepts (or sub-problems) and between groups of concepts. For example, during the hierarchy derivation process implemented in our proposed approaches, the degree to which concepts and groups of concepts are ready to be learned together is evaluated recursively until we end up with atomic concepts at the leaf nodes. The evaluated degree translates how well collaboration via transfer, sharing, and reuse can be performed across concepts and groups of concepts. In Chapter 6, collaboration was pursued by determining which components of the locally learned models should be kept locally and those that can be shared with other clients and aggregated into the central server in particular. The components that can be shared are referred to as universal components and are assumed to be abstracted from any bias that could stem from the context surrounding the data generators (or unbiased).

Additional domain knowledge that is leveraged and the way it is represented. Throughout the thesis, we explored various forms of domain models that we represented via different strategies. In Chapter 4, we leveraged sensing and transmission models of the data sources along with relevant principles such as temporal coherence or proportionality prior, which describe how the relevant properties of these data sources change over time. The way this knowledge was translated operationally into the learning process was, for example, via the diameter of the parameter space, where the rate of change of that diameter is defined by domain knowledge. In Chapter 5, we leveraged the semantics of the label space, which is used to organize the concepts to learn into appropriate structures. The

learning problems we dealt with are naturally cast as hierarchies with a relation of generalization/specialization. We investigated the use of hierarchies as a typical structure to organize the concepts. These structures are learned from data with the idea of maximizing transfer and sharing across the levels and nodes of the hierarchies. In Chapter 6, we leveraged topological models describing the disposition of the sensing devices as well as equational models describing the phenomena considered for learning. In particular, we proposed to express the geometry of the sensing devices and how they relate to each other (in terms of the views they provide) using group elements (belonging to the special Euclidean group $SE(3)$). In Chapter 4 and Chapter 6, the considered domain models were assumed to be available, while in Chapter 5, the structuring of the concepts was derived from data.

Modularity of the model's internals As we began to discuss in Chapter 2, modularity is a key enabler for the development of transparency in machine learning models. In this sense, our proposed approaches involve and promote modular aspects and, as such, open perspectives in terms of transparency in the context of distributed sensing environments. Two principles are of crucial importance for this: exhibiting components of the learning models, like portions of the decision boundaries or latent representation, and ensuring that they vary consistently with domain knowledge. In Chapter 4, modularity is materialized noticeably by the portions of the model's decision boundaries as well as the learning examples that sustain these portions. Some of these portions remain invariant while others change according, for example, to the evolution of the sensing environments or the sensing characteristics of the sensing devices. Our proposed approaches provide means for controlling in a fine-grained manner these portions of the model's decision boundaries by identifying the learning examples that are relevant w.r.t. the knowledge available about the sensing environment. In Chapter 5, modules correspond to the models (or neural networks) assigned to the nodes of the derived hierarchies. Depending on the level of the hierarchy to which a module is assigned to, different features (or biases) with varying levels of abstraction are captured. These are consequently more adapted to the concepts or groups of concepts involved in that specific level. Additionally, the hierarchical structuring of the concepts allows the emergence of inheritance mechanisms of inductive biases across the levels of the structure. In Chapter 6, modularity was explored from the perspective of learning group-invariant representations w.r.t. the different views provided by the distributed sensing environment. These correspond to the universal components that we have shown to be invariant across the clients of the sensing environment. Also, the way we ensure that each position-specific action sub-group acts precisely on specific regions of the latent space allows the emergence of modularity. Visualizations of the latent space showed the emergence of components that are activated

only by particular position-specific action sub-groups.

Structural risk minimization, the introduction of structure in the set of admissible functions, and optimization landscape. Structural risk minimization [Vap91] from Vapnik works by introducing a nested structure of subsets $S_p = \{f(x, \theta), \theta \in \Theta_p\}$, such that $S_1 \subset S_2 \subset ... \subset S_n$ and their corresponding VC-dimensions of each subset increases with inclusion. In his paper, Vapnik suggested many different ways to implement this principle. For example, the structure can be given by the architecture of the neural network, where the number of units of a given layer is monotonically increased. In this case, the subsets formed as the number of hidden units is increased introduce a structure into the admissible functions implemented by the neural network. As we discussed in the conclusive comments in Chapter 4, the way learning examples are presented to the learner, as well as how the example space is augmented in specific regions, shape the optimization landscape. This curriculum-like strategy transforms the optimization problem into smaller sub-problems of increasing difficulty, similar to continuation optimization methods. In Chapter 5, by organizing the learning process in a way that it can be decomposed into several sub-problems, there is a specific ordering that is imposed over the exploration of the hypothesis space. In Chapter 6, Restriction of the space of the hypotheses to the regions which satisfy the constraints resulting from the domain. These constraints have been expressed in the form of mathematical operators and serve to explore the space of hypotheses more efficiently. This was translated operationally into the regularization term based on representation alignment.

In terms of relevance, the approaches that we proposed allow:

- Reducing the quantities of data needed to learn;
- Improve robustness against the heterogeneity and dynamicity of structured sensing environments; Ultimately, a broader impact would be that the deployment of models in real-world environments can be facilitated. Moreover, their adaptability can be improved by simply using additional descriptions from experts in the field and describing the new deployment constraints;
- First steps towards explainable and transparent models through the incorporation of domain knowledge. Transparency aspects go hand in hand with the adaptability of models deployed in real-world environments: modularity, which is one of the principles of model transparency, is also a key element that facilitates model evolution. This brings model building (or the process of learning models) closer to the language used by domain experts to describe domain constraints. In particular, the emergence of universal and context-specific components is a first step towards this.

What is next?

In addition to the direct extensions that we proposed in the conclusive sections of our contribution chapters, we hope that this thesis will enable the development of:

- a brand new family of meta-learning approaches that integrate the underlying transformations from the domain into their internals. For example, leverage explicit representation of task-relatedness in principled ways, more featured structures, and heterogeneity isolation strategies to improve abstraction strategies;
- new federated learning approaches by explicitly leveraging client-relatedness and domain knowledge describing various aspects of the deployments. In addition to fine-grained and principled conciliation strategies to achieve better collaboration;
- a new family of approaches that optimizes simultaneously for modularity and correspondence with the domain to ultimately endow models with transparency.

In the following, we provide some open problems and interesting directions to pursue.

Conciliation phase and collaboration. Collaboration in the context of massively distributed and decentralized data is an important aspect of the learning process. As we reviewed it above, we investigated throughout this thesis various strategies for making the different parts of the distributed environments collaborate with each other to achieve a common goal, i.e., learn a unified theory. Either by making clients learn to exhibit parts of the data that can be shared with their counterparts or by constructing appropriate structures allowing a targetted transfer, sharing, and reuse across sub-groups that compose a learning problem. Unfortunately, the problem of collaboration in the massively distributed and decentralized context is challenging, with many interesting questions remaining to answer. In particular, the operational part of it, i.e., the conciliation phase responsible for aggregating what was learned locally and, by extension, the transfer, sharing, and reuse performed across sub-groups that compose a learning problem.

(i) Which elements of the locally learned models should be shared globally with others, and which should be kept locally? how to achieve this separation process? and ultimately, how to do it in a principled way? For example, like what we proposed in chapter 6 where we exploit the representation of the domain and the transformations that govern the clients' geometry in order to control the conciliation process, in particular with the emergence of universal components and others specific to the clients or to the context surrounding these clients.

(ii) How to aggregate what was learned locally? and with whom to aggregate, i.e., how to organize clients so that they collaborate together on particular aspects and potentially at different levels of abstraction? In particular, fine-grained aggregation strategies, like the FedMA algorithm [Wan+20a] (mentioned in Section 3.5) featuring a layer-wise permutation-invariant aggregation strategy, are to be privileged. Especially when there is semantics underlying the structure of the architecture of the locally learned models, e.g., position-specific and universal components that we proposed in Chapter 6. In this chapter precisely, we proposed a conciliation strategy ensuring that the relative geometry of the clients, represented in the form of group elements, acts on specific regions of the learned latent space, making it robust and flexible. More principled strategies are needed to make this process even more robust. One promising direction could be to rely on the underlying theory of Lie group and corresponding algebra. Lie groups are a special and large class of continuous groups that includes many valuable transformations like translations, rotations, and scaling, and which also proposes a principled way for handling operations on the transformations such as composition, inversion, differentiation, and interpolation. These operations are of utmost importance for representing more complex phenomena in a fine-grained manner. The idea is that they offer better parameterizations for certain problems, e.g., optimizing problems involving rotations or translations can be solved faster via a parameterization based on Lie group.

Correspondance between domain knowledge and model's parameters. Exhibiting particular parts of the learning models and ensuring that they consistently capture precise aspects from the domain is essential. This allows building robust learners that comply with the various constraints they are surrounded with and flexibly evolve when these constraints mutate. Similar in spirit to the correspondence between models of computation and proof systems established by Curry and Howard [How80], one rich avenue that is of great interest to explore is to lay down the foundations for a correspondence between the learning models (with their computational capabilities) and domain knowledge (expressed in an appropriate language). In other words, designing models that encode, within their internal mechanisms and components, inductive biases corresponding effectively to domain knowledge. Various works have been undertaken in the sense of bridging the gap between connectionist models, on the one hand, and knowledge representation and reasoning approaches, on the other hand. For example, works under the umbrella of neuro-symbolic artificial intelligence [DKT07; De +19; Hit+22]. Modularity, i.e., each component of the connectionist models is responsible for a specific piece of knowledge, is a key characteristic of such systems. Our contributions, especially [HO20] and Chapter 6, partially deal with this issue. Unfortunately, while it is a key enabler for our target framework, building this correspondence remains

a challenging open problem that needs careful attention.

A possible avenue that could be explored would be to express domain knowledge represented in the form of group-invariance/equivariance and integrate it into the learning models via parameter-sharing schemes as outlined by [RSP17]: given a group G (corresponding to available domain knowledge) that acts discretely on the input and output of a standard neural network layer, the weights of the layer are equivariant with respect to G-action iff G explains the symmetries of the network parameters. In other words, priors on the input/output structure of neural networks can be encoded through parameter-sharing. Indeed, parameter-sharing is concerned with how the nodes of the neural networks are linked to each other (not to the values of the weights which link them) and how these specific parameter-sharing patterns encode specific domain invariants and equivariants. For example, convolutional neural networks implement inductive biases (local sensitivity, invariance, etc.) via the notion of parameter-sharing concretely implemented by the convolution operation. This kind of strategy bears a resemblance to neural architecture search approaches (weight-agnostic neural networks [GH19], lottery hypothesis [FC18], etc.), which optimize for the most appropriate tying of the neurons (not the weights themselves). The weight-tying (or parameter-sharing) scheme is what defines an architecture and, by extension, an inductive bias. Similarly, Transformers [Vas+17] can be explored for their flexibility in terms of parameter-tying. Indeed, transformers are general computing machines that do not have fixed inductive biases and instead, in some sense, learn appropriate inductive biases. The mechanism underlying transformers is attention. It is a quadratic operation that tries to make its components (tokens) attend to the entire input sequences and also attend to each other. Ultimately, this process tries to create specific relations by tying specific tokens to each other.

How domain knowledge shapes exactly the optimization landscape. We have seen throughout this thesis that the way we act on the learning process by integrating domain knowledge has a substantial impact on the optimization landscape. It has the notable ability to shape the optimization landscape in ways that facilitate its exploration. For example, we leveraged in Chapter 6 invariants from the domain in the form of symmetries between the different views provided by the distributed sensing devices in order to constrain the learning process. As we discussed previously, summarized as: "symmetries of the observation models become symmetries of the optimization problem" [ZQW20], symmetries of the domain are widely studied in terms of the impact that they induce on the optimization landscape. For example, as outlined in [Li+19b], the rotational symmetry group induces many nonisolated saddle points and equivalent global minima. Similarly, the way the learning process in Chapter 5 is organized into structured sets of sub-problems that interact with each other via transfer, sharing, and reuse, also

translates into particular forms of the optimization landscape with potentially interesting properties. These potential properties could be leveraged for efficient traversal of the optimization landscape.

Characterization of the optimization landscape in the presence of domain invariants. How does domain knowledge shape the optimization landscape? which properties do invariants from the domain bring to this landscape? One axe of development could be the characterization of the links between domain models (like invariants) with the optimization landscape. The idea is to exhibit interesting properties that could be leveraged for better traversal of the optimization landscape.

Inductive leaps in the optimization space. How can we leverage the optimization landscape's properties to make substantial inductive leaps? and accelerate convergence toward good solutions? One perspective regarding the traversal of the optimization landscape is to devise and make use of global optimization methods instead of local ones. The idea is to leverage the potential properties of the optimization landscape shaped by domain knowledge to make substantial inductive leaps. As we discussed in the conclusive comments of Chapter 4, a challenge in this regard is that an adjustment in the parameters does not necessarily translate into a change in the function space.

Optimization on Riemannian manifolds. As proposed above, using more principled strategies to express the relative geometry of data sources, e.g., in the form of Lie groups, generalizes the learning problem to manifolds, requiring optimization with respect to parameters in curved spaces [TO20]. Indeed, another promising avenue to pursue is the optimization on smooth or Riemannian manifolds, which boils down basically to optimizing on a known manifold structure [RW12]. This has the potential to accelerate the optimization process as there is a vast literature on optimization methods that are more adapted to these manifolds, e.g., the classical convergence results and algorithms from the Euclidean setting have been adapted to the Riemannian one [AMS09; Bon13; BAC19; ZJS16; Liu+17; SKM19; LM19]. Similarly, smooth optimization of functions on Lie groups, e.g., [TO20], can be pursued as many aspects from the domain can be encoded in these kinds of mathematical operators.

Bibliography

[Aba+16] Martin Abadi et al. "Tensorflow: a system for large-scale machine learning." In: *OSDI*. Vol. 16. 2016, pp. 265–283 (cit. on p. 179).

[Abe18] Mohamed Tayeb AbedGhars. "Contribution à la caractérisation et synthèse de pigment de peinture à base de calamine. Evaluation de la qualité et analyse des incertitudes sur les propriétés." PhD thesis. Université Badji Mokhtar de Annaba, 2018 (cit. on p. 117).

[ABG17] Mohamed Tayeb AbedGhars, Salah Bouhouche, and Mokhtar Ghers. "Prediction of thermal and mass loss behavior of mineral mixture using inferential stochastic modeling and thermal analysis measurement data". In: *Measurement* 109 (2017), pp. 326–333 (cit. on p. 117).

[AC09] Hamid Aghajan and Andrea Cavallaro. *Multi-camera networks: principles and applications*. Academic press, 2009 (cit. on pp. 15, 81, 132, 173).

[Aca+20] Durmus Alp Emre Acar et al. "Federated Learning Based on Dynamic Regularization". In: *International Conference on Learning Representations*. 2020 (cit. on pp. 75, 77).

[ACB17] Martin Arjovsky, Soumith Chintala, and Léon Bottou. "Wasserstein generative adversarial networks". In: *International conference on machine learning*. PMLR. 2017, pp. 214–223 (cit. on p. 111).

[Ach+19] Alessandro Achille et al. "Task2vec: Task embedding for meta-learning". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2019, pp. 6430–6439 (cit. on pp. 61, 64, 136).

[AES19] Antreas Antoniou, Harri Edwards, and Amos Storkey. "How to train your MAML". In: *Seventh International Conference on Learning Representations*. 2019 (cit. on pp. 46, 54).

[AG12] Eugene L Allgower and Kurt Georg. *Numerical continuation methods: an introduction*. Vol. 13. Springer Science & Business Media, 2012 (cit. on pp. 83, 126).

[Ahm+19] Zafarali Ahmed et al. "Understanding the impact of entropy on policy optimization". In: *International conference on machine learning*. PMLR. 2019, pp. 151–160 (cit. on p. 127).

[Alb+08] Alhussein Albarbar et al. "Suitability of MEMS accelerometers for condition monitoring: An experimental study". In: *Sensors* 8.2 (2008), pp. 784–799 (cit. on p. 7).

[Alj+19] Rahaf Aljundi et al. "Gradient based sample selection for online continual learning". In: *Advances in neural information processing systems* 32 (2019) (cit. on pp. 84, 85, 92).

[AMK20] Ammar Ahmad Tarar, Umair Mohammad, and Soumya K Srivastava. "Wearable skin sensors and their challenges: A review of transdermal, optical, and Mechanical Sensors". In: *Biosensors* 10.6 (2020), p. 56 (cit. on p. 12).

[AMS09] P-A Absil, Robert Mahony, and Rodolphe Sepulchre. "Optimization algorithms on matrix manifolds". In: *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, 2009 (cit. on p. 206).

[And+13] Galen Andrew et al. "Deep canonical correlation analysis". In: *International conference on machine learning*. PMLR. 2013, pp. 1247–1255 (cit. on pp. 176, 179, 192).

[Anz+20] Arman Anzanpour et al. "Edge-assisted control for healthcare internet of things: A case study on ppg-based early warning score". In: *ACM Transactions on Internet of Things* 2.1 (2020), pp. 1–21 (cit. on pp. 11, 13).

[Asi+20] Yusra Asim et al. "Context-Aware Human Activity Recognition (CAHAR) in-the-Wild Using Smartphone Accelerometer". In: *IEEE Sensors Journal* 20.8 (2020), pp. 4361–4371 (cit. on p. 172).

[Ata+11] Louis Atallah et al. "Sensor positioning for activity recognition using wearable accelerometers". In: *IEEE transactions on biomedical circuits and systems* 5.4 (2011), pp. 320–329 (cit. on p. 15).

[Att+15] Ferhat Attal et al. "Physical human activity recognition using wearable sensors". In: *Sensors* 15.12 (2015), pp. 31314–31338 (cit. on p. 15).

[Ayk+19] Muratahan Aykol et al. "Network analysis of synthesizable materials discovery". In: *Nature communications* 10.1 (2019), p. 2018 (cit. on p. 27).

[AYS21] Mayank Agarwal, Mikhail Yurochkin, and Yuekai Sun. "On sensitivity of meta-learning to support data". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 20447–20460 (cit. on p. 58).

[BAC19] Nicolas Boumal, Pierre-Antoine Absil, and Coralia Cartis. "Global rates of convergence for nonconvex optimization on manifolds". In: *IMA Journal of Numerical Analysis* 39.1 (2019), pp. 1–33 (cit. on p. 206).

[Bak+17] Bowen Baker et al. "Accelerating neural architecture search using performance prediction". In: *arXiv preprint arXiv:1705.10823* (2017) (cit. on p. 59).

[Ban+14] Oresti Banos et al. "Dealing with the effects of sensor displacement in wearable activity recognition". In: *Sensors* 14.6 (2014), pp. 9995–10023 (cit. on p. 171).

[Ban12] Ayan Banerjee. *Model based safety analysis and verification of cyber-physical systems*. Arizona State University, 2012 (cit. on pp. 11, 13, 14).

[Bau+17] David Bau et al. "Network dissection: Quantifying interpretability of deep visual representations". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 6541–6549 (cit. on p. 56).

[Bax00] Jonathan Baxter. "A model of inductive bias learning". In: *Journal of artificial intelligence research* 12 (2000), pp. 149–198 (cit. on pp. 33, 34, 36, 44, 48, 61, 62, 87).

[BB08] Shai Ben-David and Reba Schuller Borbely. "A notion of task relatedness yielding provable multiple-task learning guarantees". In: *Machine learning* 73.3 (2008), pp. 273–287 (cit. on p. 62).

[BBC91] Y Bengio, S Bengio, and J Cloutier. "Learning a synaptic learning rule". In: *IJCNN-91-Seattle International Joint Conference on Neural Networks*. Vol. 2. IEEE. 1991, 969–vol (cit. on pp. 4, 20).

[BBS14] Andreas Bulling, Ulf Blanke, and Bernt Schiele. "A tutorial on human activity recognition using body-worn inertial sensors". In: *ACM Computing Surveys (CSUR)* (2014) (cit. on pp. 28, 96, 168, 171).

[Bec+21] Sarah Bechtle et al. "Meta learning via learned loss". In: *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE. 2021, pp. 4161–4168 (cit. on pp. 21, 35, 42).

[Bel+17] Irwan Bello et al. "Neural optimizer search with reinforcement learning". In: *International Conference on Machine Learning*. PMLR. 2017, pp. 459–468 (cit. on pp. 20, 35, 40, 41).

[Ben+09] Yoshua Bengio et al. "Curriculum learning". In: *Proceedings of the 26th annual international conference on machine learning*. 2009, pp. 41–48 (cit. on pp. 83, 94, 126).

[Ben+10] Shai Ben-David et al. "A theory of learning from different domains". In: *Machine learning* 79.1 (2010), pp. 151–175 (cit. on p. 62).

[Ber+20] Jeremy Bernstein et al. "On the distance between two neural networks and the stability of learning". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 21370–21381 (cit. on p. 129).

[Bha+20] Ganapati Bhat et al. "w-HAR: An activity recognition dataset and framework using low-power wearable devices". In: *Sensors* 20.18 (2020), p. 5356 (cit. on p. 9).

[BHS13] Aurélien Bellet, Amaury Habrard, and Marc Sebban. "A survey on metric learning for feature vectors and structured data". In: *arXiv preprint arXiv:1306.6709* (2013) (cit. on p. 43).

[Bis06] Christopher M Bishop. "Pattern Recognition and Machine Learning". In: (2006) (cit. on pp. 31, 91).

[BK12] Wei Bi and James Tin-Yau Kwok. "Mandatory leaf node prediction in hierarchical multilabel classification". In: *Advances in Neural Information Processing Systems* 1 (2012), p. 153 (cit. on p. 143).

[BKT19] Maria-Florina Balcan, Mikhail Khodak, and Ameet Talwalkar. "Provable guarantees for gradient-based meta-learning". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 424–433 (cit. on p. 87).

[Blu+89] Anselm Blumer et al. "Learnability and the Vapnik-Chervonenkis dimension". In: *Journal of the ACM (JACM)* 36.4 (1989), pp. 929–965 (cit. on pp. 19, 32).

[Bol+17] Eliot Bolduc et al. "Projected gradient descent algorithms for quantum state tomography". In: *npj Quantum Information* 3.1 (2017), pp. 1–9 (cit. on p. 116).

[Bon13] Silvere Bonnabel. "Stochastic gradient descent on Riemannian manifolds". In: *IEEE Transactions on Automatic Control* 58.9 (2013), pp. 2217–2229 (cit. on p. 206).

[Bor] Murmann Boris. *ADC Performance Survey 1997-2017*. http://web.stanford.edu/~murmann/adcsurvey.html (cit. on p. 6).

[BP95] David Beymer and Tomaso Poggio. “Face recognition from one example view”. In: *Proceedings of IEEE International Conference on Computer Vision*. IEEE. 1995, pp. 500–507 (cit. on p. 104).

[Bra+22] Pavel Brazdil et al. “Dataset Characteristics (Metafeatures)”. In: *Metalearning*. Springer, 2022, pp. 53–75 (cit. on p. 60).

[Bre96] Leo Breiman. “Bagging predictors”. In: *Machine learning* 24.2 (1996), pp. 123–140 (cit. on p. 38).

[BRK18] Ari Benjamin, David Rolnick, and Konrad Kording. “Measuring and regularizing networks in function space”. In: *International Conference on Learning Representations*. 2018 (cit. on p. 129).

[BV04] Stephen P Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004 (cit. on pp. 34, 116).

[BY20] Billur Barshan and Aras Yurtman. “Classifying Daily and Sports Activities Invariantly to the Positioning of Wearable Motion Sensor Units”. In: *IEEE Internet of Things Journal* (2020) (cit. on pp. 168, 171).

[Cal+18] Sebastian Caldas et al. “Leaf: A benchmark for federated settings”. In: *arXiv preprint arXiv:1812.01097* (2018) (cit. on pp. 79, 86).

[Car+18] James J Carollo et al. “Relative phase measures of intersegmental coordination describe motor control impairments in children with cerebral palsy who exhibit stiff-knee gait”. In: *Clinical Biomechanics* 59 (2018), pp. 40–46 (cit. on pp. 168, 169, 173).

[CD15] Mauri´cio de Campos Porath and Ricardo Dolci. “Uncertainty of angular displacement measurement with a MEMS gyroscope integrated in a smartphone”. In: *J. Phys. Conf. Ser.* Vol. 648. 1. 2015, p. 012007 (cit. on p. 7).

[CGF19] Hugo Caselles-Dupré, Michael Garcia Ortiz, and David Filliat. “Symmetry-based disentangled representation learning requires interaction with environments”. In: *Advances in Neural Information Processing Systems* 32 (2019), pp. 4606–4615 (cit. on p. 170).

[CGW20] Wuyang Chen, Xinyu Gong, and Zhangyang Wang. “Neural Architecture Search on ImageNet in Four GPU Hours: A Theoretically Inspired Perspective”. In: *International Conference on Learning Representations*. 2020 (cit. on p. 51).

[CH04] Lijuan Cai and Thomas Hofmann. "Hierarchical document categorization with support vector machines". In: *Proceedings of the thirteenth ACM international conference on Information and knowledge management*. 2004, pp. 78–87 (cit. on p. 132).

[Che+18] Fei Chen et al. "Federated meta-learning for recommendation. CoRR abs/1802.07876 (2018)". In: *arXiv preprint arXiv:1802.07876* (2018) (cit. on p. 80).

[Che+19] Yutian Chen et al. "Modular meta-learning with shrinkage". In: *arXiv preprint arXiv:1909.05557* (2019) (cit. on pp. 46, 53, 55, 56).

[Cho+14] Kyunghyun Cho et al. "On the properties of neural machine translation: Encoder-decoder approaches". In: *arXiv preprint arXiv:1409.1259* (2014) (cit. on p. 39).

[CK21] Zachary Charles and Jakub Konečnỳ. "Convergence and accuracy trade-offs in federated learning and meta-learning". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2021, pp. 2575–2583 (cit. on pp. 79, 80).

[Coh60] Jacob Cohen. "A coefficient of agreement for nominal scales". In: *Educational and psychological measurement* 20.1 (1960), pp. 37–46 (cit. on p. 162).

[Coo+12] Fallon Cook et al. "Baby Business: a randomised controlled trial of a universal parenting program that aims to prevent early infant sleep and cry problems and associated parental depression". In: *BMC pediatrics* 12.1 (2012), p. 13 (cit. on p. 27).

[Cor07] A Cornuejols. "The necessity of order in machine learning: Is order in order". In: *In Order to Learn: How the Sequence of Topics Influences Learning* 2 (2007), p. 41 (cit. on p. 127).

[CPC19] Diogo V Carvalho, Eduardo M Pereira, and Jaime S Cardoso. "Machine learning interpretability: A survey on methods and metrics". In: *Electronics* 8.8 (2019), p. 832 (cit. on p. 2).

[Cub+19] Ekin D Cubuk et al. "Autoaugment: Learning augmentation strategies from data". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 113–123 (cit. on pp. 103, 104).

[CV95] Corinna Cortes and Vladimir Vapnik. "Support-vector networks". In: *Machine learning* 20.3 (1995), pp. 273–297 (cit. on p. 91).

[Dan+18] Loai Danial et al. "Breaking through the speed-power-accuracy tradeoff in ADCs using a memristive neuromorphic architecture". In: *IEEE Transactions on Emerging Topics in Computational Intelligence* 2.5 (2018), pp. 396–409 (cit. on p. 6).

[Dau+14] Yann N Dauphin et al. "Identifying and attacking the saddle point problem in high-dimensional non-convex optimization". In: *Advances in neural information processing systems* 27 (2014) (cit. on pp. 127, 128).

[DB17] Emily L Denton and Vighnesh Birodkar. "Unsupervised Learning of Disentangled Representations from Video". In: *NIPS*. 2017 (cit. on p. 175).

[DBJ22] Yatin Dandi, Luis Barba, and Martin Jaggi. "Implicit gradient alignment in distributed and federated learning". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 36. 6. 2022, pp. 6454–6462 (cit. on p. 77).

[De +19] Luc De Raedt et al. "Neuro-symbolic= neural+ logical+ probabilistic". In: *NeSy'19@ IJCAI, the 14th International Workshop on Neural-Symbolic Learning and Reasoning*. 2019 (cit. on pp. 23, 204).

[De +21] Matthias De Lange et al. "A continual learning survey: Defying forgetting in classification tasks". In: *IEEE transactions on pattern analysis and machine intelligence* 44.7 (2021), pp. 3366–3385 (cit. on p. 85).

[Dey+14] Sanorita Dey et al. "AccelPrint: Imperfections of Accelerometers Make Smartphones Trackable." In: *NDSS*. Citeseer. 2014 (cit. on pp. 8, 68).

[DHS11] John Duchi, Elad Hazan, and Yoram Singer. "Adaptive subgradient methods for online learning and stochastic optimization." In: *Journal of machine learning research* 12.7 (2011) (cit. on p. 75).

[DKT07] Luc De Raedt, Angelika Kimmig, and Hannu Toivonen. "ProbLog: A Probabilistic Prolog and Its Application in Link Discovery." In: *IJCAI*. Vol. 7. Hyderabad. 2007, pp. 2462–2467 (cit. on pp. 23, 204).

[DO09] Raffaele D'Errico and Laurent Ouvry. "Time-variant BAN channel characterization". In: *2009 IEEE 20th International Symposium on Personal, Indoor and Mobile Radio Communications*. IEEE. 2009, pp. 3000–3004 (cit. on p. 9).

[Du+20] Simon Shaolei Du et al. "Few-Shot Learning via Learning the Representation, Provably". In: *International Conference on Learning Representations*. 2020 (cit. on p. 62).

[Dun09] Priscilla Dunstan. *Child Sense: From Birth to Age 5, how to Use the 5 Senses to Make Sleeping, Eating, Dressing, and Other Everyday Activities Easier While Strengthening Your Bond with Your Child*. Bantam, 2009 (cit. on p. 27).

[Eha+20] Muhammad Ehatisham-Ul-Haq et al. "C2FHAR: Coarse-to-Fine Human Activity Recognition With Behavioral Context Modeling Using Smart Inertial Sensors". In: *IEEE Access* 8 (2020), pp. 7731–7747 (cit. on p. 172).

[EOR15] Moez Essaidi, Aomar Osmani, and Céline Rouveirol. "Learning Dependent-Concepts in ILP: Application to Model-Driven Data Warehouses". In: *Latest Advances In Inductive Logic Programming*. World Scientific, 2015, pp. 151–172 (cit. on p. 132).

[EP04] Theodoros Evgeniou and Massimiliano Pontil. "Regularized multi–task learning". In: *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. 2004, pp. 109–117 (cit. on p. 63).

[ES16] Harrison Edwards and Amos Storkey. "Towards a neural statistician". In: (2016) (cit. on p. 60).

[ESS19] Riham Elhabyan, Wei Shi, and Marc St-Hilaire. "Coverage protocols for wireless sensor networks: Review and future directions". In: *Journal of Communications and Networks* 21.1 (2019), pp. 45–60 (cit. on p. 102).

[Est+19] Carlos Esteves et al. "Equivariant multi-view networks". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 1568–1577 (cit. on p. 173).

[Eur17] European parilament. *Regulation (EU) 2017/745 of the European Parliament and of the Council of 5 April 2017 on medical devices, amending Directive 2001/83/EC, Regulation (EC) No 178/2002 and Regulation (EC) No 1223/2009 and repealing Council Directives 90/385/EEC and 93/42/EEC*. 2017. URL: http://data.europa.eu/eli/reg/2017/745/oj (cit. on p. 16).

[Fai16] Paul Fairfield. *Teachability and Learnability: Can Thinking be Taught?* Routledge, 2016 (cit. on p. 126).

[FAL17] Chelsea Finn, Pieter Abbeel, and Sergey Levine. "Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks". In: *ICML*. 2017 (cit. on pp. 20, 32, 35, 44, 45, 47, 85).

[Fan+18] Yang Fan et al. "Learning to Teach". In: *International Conference on Learning Representations*. 2018 (cit. on pp. 83, 126).

[FB17] C Daniel Freeman and Joan Bruna. "Topology and geometry of half-rectified network optimization". In: *5th International Conference on Learning Representations, ICLR 2017*. 2017 (cit. on p. 34).

[FC18] Jonathan Frankle and Michael Carbin. "The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks". In: *International Conference on Learning Representations*. 2018 (cit. on pp. 129, 149, 205).

[FGH06] Peter H Feiler, David P Gluch, and John J Hudak. *The architecture analysis & design language (AADL): An introduction*. Tech. rep. Carnegie-Mellon Univ Pittsburgh PA Software Engineering Inst, 2006 (cit. on p. 13).

[Fin+20] Marc Finzi et al. "Generalizing convolutional neural networks for equivariance to lie groups on arbitrary continuous data". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 3165–3176 (cit. on p. 170).

[Fin18] Chelsea B Finn. "Learning to Learn with Gradients". PhD thesis. University of California, Berkeley, 2018 (cit. on pp. 4, 20, 22, 60).

[Fle+19] Sebastian Flennerhag et al. "Meta-Learning with Warped Gradient Descent". In: *International Conference on Learning Representations*. 2019 (cit. on pp. 47, 48).

[Fle21] Sebastian Flennerhag. "Towards Scalable Meta-Learning". PhD thesis. The University of Manchester (United Kingdom), 2021 (cit. on pp. 34, 48).

[FMO20] Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. "Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 3557–3568 (cit. on p. 80).

[For+05] Andrew Fort et al. "Characterization of the ultra wideband body area propagation channel". In: *2005 IEEE International Conference on Ultra-Wideband*. IEEE. 2005, 6–pp (cit. on pp. 9, 10).

[Fra+18] Luca Franceschi et al. "Bilevel programming for hyperparameter optimization and meta-learning". In: *International Conference on Machine Learning*. PMLR. 2018, pp. 1568–1577 (cit. on p. 38).

[FS10] George Forman and Martin Scholz. "Apples-to-apples in cross-validation studies: pitfalls in classifier performance measurement". In: *ACM SIGKDD Explorations Newsletter* 12.1 (2010), pp. 49–57 (cit. on p. 192).

[Ge+15] Rong Ge et al. “Escaping from saddle points—online stochastic gradient for tensor decomposition”. In: *Conference on learning theory*. PMLR. 2015, pp. 797–842 (cit. on p. 128).

[GFG20] Micah Goldblum, Liam Fowl, and Tom Goldstein. “Adversarially robust few-shot learning: A meta-learning approach”. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 17886–17895 (cit. on p. 58).

[GG11] Hristijan Gjoreski and Matjaž Gams. “Accelerometer data preparation for activity recognition”. In: *Proceedings of the International Multiconference Information Society, Ljubljana, Slovenia*. Vol. 1014. 2011, p. 1014 (cit. on p. 15).

[GH19] Adam Gaier and David Ha. “Weight agnostic neural networks”. In: *Advances in Neural Information Processing Systems*. 2019, pp. 5364–5378 (cit. on pp. 51, 52, 104, 129, 149, 205).

[Gil+18] Leilani H Gilpin et al. “Explaining explanations: An overview of interpretability of machine learning”. In: *2018 IEEE 5th International Conference on data science and advanced analytics (DSAA)*. IEEE. 2018, pp. 80–89 (cit. on pp. 2, 56).

[Gjo+18] Hristijan Gjoreski et al. “The University of Sussex-Huawei locomotion and transportation dataset for multimodal analytics with mobile devices”. In: *IEEE Access* (2018) (cit. on pp. 71, 94, 135, 191).

[GL21] Jiechao Guan and Zhiwu Lu. “Task Relatedness-Based Generalization Bounds for Meta Learning”. In: *International Conference on Learning Representations*. 2021 (cit. on p. 87).

[GM17] Rong Ge and Tengyu Ma. “On the optimization landscape of tensor decompositions”. In: *Advances in Neural Information Processing Systems* 30 (2017) (cit. on p. 128).

[Gol+20] Micah Goldblum et al. “Unraveling meta-learning: Understanding feature representations for few-shot tasks”. In: *International Conference on Machine Learning*. PMLR. 2020, pp. 3607–3616 (cit. on pp. 50, 51).

[Gol05] Andrea Goldsmith. “Path Loss and Shadowing”. In: *Wireless Communications*. Cambridge University Press, 2005, pp. 27–63. DOI: 10.1017/CBO9780511841224.003 (cit. on p. 9).

[Gol85] Michael Goldstein. “Temporal coherence”. In: *Bayesian Statistics* 2 (1985), pp. 231–248 (cit. on p. 89).

[Goo+14] Ian Goodfellow et al. "Generative adversarial nets". In: *Advances in neural information processing systems* 27 (2014) (cit. on p. 105).

[Gor+09] Jean-Marie Gorce et al. "Opportunistic relaying protocols for human monitoring in BAN". In: *2009 IEEE 20th International Symposium on Personal, Indoor and Mobile Radio Communications*. IEEE. 2009, pp. 732–736 (cit. on p. 9).

[GPW19] Badih Ghazi, Rina Panigrahy, and Joshua Wang. "Recursive sketches for modular deep learning". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 2211–2220 (cit. on pp. 56, 57).

[Gri+10] Thomas L Griffiths et al. "Probabilistic models of cognition: Exploring representations and inductive biases". In: *Trends in cognitive sciences* 14.8 (2010), pp. 357–364 (cit. on p. 35).

[GYP20] Gunshi Gupta, Karmesh Yadav, and Liam Paull. "Look-ahead meta learning for continual learning". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 11588–11598 (cit. on p. 55).

[Han+21] Jindong Han et al. "Joint air quality and weather prediction based on multi-adversarial spatiotemporal networks". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 5. 2021, pp. 4081–4089 (cit. on p. 2).

[Hat+20] Ryuichiro Hataya et al. "Faster autoaugment: Learning augmentation strategies using backpropagation". In: *European Conference on Computer Vision*. Springer. 2020, pp. 1–16 (cit. on p. 104).

[He+20] Chaoyang He et al. "Fedml: A research library and benchmark for federated machine learning". In: *arXiv preprint arXiv:2007.13518* (2020) (cit. on p. 70).

[Heo+19] Byeongho Heo et al. "Knowledge distillation with adversarial samples supporting decision boundary". In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 33. 01. 2019, pp. 3771–3778 (cit. on pp. 91, 93, 94).

[HHL14] Frank Hutter, Holger Hoos, and Kevin Leyton-Brown. "An efficient approach for assessing hyperparameter importance". In: *International conference on machine learning*. PMLR. 2014, pp. 754–762 (cit. on p. 150).

[Hig+17] Irina Higgins et al. "beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework". In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL: https://openreview.net/forum?id=Sy2fzU9gl (cit. on p. 175).

[Hig+18] Irina Higgins et al. "Towards a definition of disentangled representations". In: *arXiv preprint arXiv:1812.02230* (2018) (cit. on p. 190).

[Hit+22] Pascal Hitzler et al. "Neuro-symbolic approaches in artificial intelligence". In: *National Science Review* 9.6 (2022), nwac035 (cit. on p. 204).

[HKV19] Frank Hutter, Lars Kotthoff, and Joaquin Vanschoren. *Automated machine learning: methods, systems, challenges*. Springer Nature, 2019 (cit. on p. 42).

[HO20] Massinissa Hamidi and Aomar Osmani. "Data Generation Process Modeling for Activity Recognition". In: *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*. Springer. 2020 (cit. on pp. 15, 25, 29, 56, 57, 81, 83, 132, 150, 168, 173, 204).

[HO21a] Massinissa Hamidi and Aomar Osmani. "Domain models for data sources integration in HAR". In: *Neurocomputing* 444 (2021), pp. 244–259 (cit. on p. 30).

[HO21b] Massinissa Hamidi and Aomar Osmani. "Human Activity Recognition: A Dynamic Inductive Bias Selection Perspective". In: *Sensors* 21.21 (2021), p. 7278 (cit. on pp. 5, 28, 30, 61, 68, 168, 171).

[HO22] Massinissa Hamidi and Aomar Osmani. "Context Abstraction to Improve Decentralized Machine Learning in Structured Sensing Environments". In: *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*. Springer. 2022 (cit. on pp. 26, 28, 76, 167).

[HO23] Massinissa Hamidi and Aomar Osmani. "Learning to Select Learning Examples in Structured Decentralized Sensing Environments". In: *Submitted*. 2023 (cit. on pp. 25, 83).

[HOA20a] Massinissa Hamidi, Aomar Osmani, and Pegah Alizadeh. "A Multi-View Architecture for the SHL Challenge". In: UbiComp-ISWC '20. New York, NY, USA: Association for Computing Machinery, 2020, pp. 317–322 (cit. on pp. 29, 173).

[HOA20b] Massinissa Hamidi, Aomar Osmani, and Pegah Alizadeh. "A Multi-View Architecture for the SHL Challenge". In: UbiComp-ISWC '20. New York, NY, USA: Association for Computing Machinery, 2020, pp. 317–322 (cit. on p. 158).

[Hos+21] Timothy M Hospedales et al. "Meta-Learning in Neural Networks: A Survey". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021) (cit. on pp. 32, 37, 42, 58).

[Hot92] Harold Hotelling. "Relations between two sets of variates". In: *Breakthroughs in statistics*. Springer, 1992, pp. 162–190 (cit. on p. 177).

[How80] William A Howard. "The formulae-as-types notion of construction". In: *To HB Curry: essays on combinatory logic, lambda calculus and formalism* 44 (1980), pp. 479–490 (cit. on p. 204).

[HP15] Nils Y Hammerla and Thomas Plötz. "Let's (not) stick together: pairwise similarity biases cross-validation in activity recognition". In: *Proceedings of the 2015 ACM international joint conference on pervasive and ubiquitous computing*. 2015, pp. 1041–1051 (cit. on pp. 42, 43, 145, 146, 192).

[HQB19] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. "Measuring the effects of non-identical data distribution for federated visual classification". In: *arXiv preprint arXiv:1909.06335* (2019) (cit. on p. 75).

[HR09] Niall Hurley and Scott Rickard. "Comparing measures of sparsity". In: *IEEE Transactions on Information Theory* 55.10 (2009), pp. 4723–4741 (cit. on p. 180).

[HRP21] Mike Huisman, Jan N van Rijn, and Aske Plaat. "A survey of deep meta-learning". In: *Artificial Intelligence Review* (2021), pp. 1–59 (cit. on p. 43).

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory". In: *Neural computation* 9.8 (1997), pp. 1735–1780 (cit. on pp. 39, 179).

[Hsi+17] Kevin Hsieh et al. "Gaia: Geo-distributed machine learning approaching {LAN} speeds". In: *14th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 17)*. 2017, pp. 629–647 (cit. on p. 3).

[Hsi+18] Jun-Ting Hsieh et al. "Learning to decompose and disentangle representations for video prediction". In: *arXiv preprint arXiv:1806.04166* (2018) (cit. on p. 175).

[Hsi+20] Kevin Hsieh et al. "The non-iid data quagmire of decentralized machine learning". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 4387–4398 (cit. on pp. 3, 70, 171).

[HSS12] Geoffrey Hinton, Nitish Srivastava, and Kevin Swersky. "Neural networks for machine learning lecture 6a overview of mini-batch gradient descent". In: *Cited on* (2012), p. 14 (cit. on p. 40).

[HVD15] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. "Distilling the knowledge in a neural network". In: *arXiv preprint arXiv:1503.02531* (2015) (cit. on pp. 58, 59, 91, 142).

[HW19] Guy Hacohen and Daphna Weinshall. "On the power of curriculum learning in training deep networks". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 2535–2544 (cit. on pp. 83, 89).

[Ida14] Nathan Ida. *Sensors, actuators, and their interfaces: A multidisciplinary introduction*. Vol. 2. IET, 2014 (cit. on p. 5).

[ISO+16] ISO13485 ISO et al. *ISO13485: 2016*. 2016 (cit. on p. 16).

[Jan+17] Majid Janidarmian et al. "A comprehensive analysis on wearable acceleration sensors in human activity recognition". In: *Sensors* 17.3 (2017), p. 529 (cit. on pp. 8, 71).

[Jer+19] Ghassen Jerfel et al. "Reconciling meta-learning and continual learning with online mixtures of tasks". In: *Advances in Neural Information Processing Systems* 32 (2019) (cit. on pp. 44, 63, 64, 136).

[Jia+18] Xiang Jiang et al. "Learning to learn with conditional class dependencies". In: *International Conference on Learning Representations*. 2018 (cit. on pp. 61, 136).

[Jia+19] Yihan Jiang et al. "Improving federated learning personalization via model agnostic meta learning". In: *arXiv preprint arXiv:1909.12488* (2019) (cit. on pp. 79, 80).

[Joh02] Steven Johnson. *Emergence: The connected lives of ants, brains, cities, and software*. Simon and Schuster, 2002 (cit. on p. 4).

[JS21] Sharu Theresa Jose and Osvaldo Simeone. "An information-theoretic analysis of the impact of task similarity on meta-learning". In: *IEEE International Symposium on Information Theory (ISIT)*. 2021 (cit. on p. 62).

[Jui15] Patrick Juignet. "Edgar Morin et la complexité [Edgar Morin and complexity]". In: *Philosophie, science et société* (2015) (cit. on p. 4).

[JVB08] Laurent Jacob, Jean-philippe Vert, and Francis Bach. "Clustered multi-task learning: A convex formulation". In: *Advances in neural information processing systems* 21 (2008) (cit. on p. 63).

[Kai+19] Peter Kairouz et al. "Advances and open problems in federated learning". In: *arXiv preprint arXiv:1912.04977* (2019) (cit. on pp. 32, 67, 69–71, 80, 171).

[Kar+06] Dean M Karantonis et al. "Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring". In: *IEEE transactions on information technology in biomedicine* 10.1 (2006), pp. 156–167 (cit. on p. 15).

[Kar+16] Maximilian Karl et al. "Deep variational bayes filters: Unsupervised learning of state space models from raw data". In: *arXiv preprint arXiv:1605.06432* (2016) (cit. on p. 197).

[Kar+20] Sai Praneeth Karimireddy et al. "Scaffold: Stochastic controlled averaging for federated learning". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 5132–5143 (cit. on pp. 74, 75, 77, 78, 194).

[Kau20] Rituraj Kaushik. "Data-Efficient Robot Learning using Priors from Simulators". PhD thesis. Université de Lorraine, 2020 (cit. on p. 62).

[Kaw16] Kenji Kawaguchi. "Deep learning without poor local minima". In: *Advances in neural information processing systems* 29 (2016) (cit. on p. 128).

[KB14] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014) (cit. on pp. 40, 75, 119).

[KBT19] Mikhail Khodak, Maria-Florina F Balcan, and Ameet S Talwalkar. "Adaptive gradient-based meta-learning methods". In: *Advances in Neural Information Processing Systems* 32 (2019) (cit. on pp. 21, 35, 53–55, 63, 64, 80, 86).

[Kim+18] Been Kim et al. "Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav)". In: *International conference on machine learning*. PMLR. 2018, pp. 2668–2677 (cit. on p. 56).

[Kim+22] Siwon Kim et al. "Towards a rigorous evaluation of time-series anomaly detection". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 36. 7. 2022, pp. 7194–7201 (cit. on p. 2).

[KKB18] Louis Kirsch, Julius Kunze, and David Barber. "Modular networks: Learning to decompose neural computation". In: *Advances in neural information processing systems* 31 (2018) (cit. on pp. 53, 55, 56).

[KL08] Kai Kunze and Paul Lukowicz. "Dealing with sensor displacement in motion-based onbody activity recognition systems". In: *Proceedings of the 10th international conference on Ubiquitous computing*. 2008, pp. 20–29 (cit. on p. 172).

[KMR15] Jakub Konečnỳ, Brendan McMahan, and Daniel Ramage. "Federated optimization: Distributed optimization beyond the datacenter". In: *arXiv preprint arXiv:1511.03575* (2015) (cit. on p. 65).

[KMR20] Ahmed Khaled, Konstantin Mishchenko, and Peter Richtárik. "Tighter theory for local SGD on identical and heterogeneous data". In: *AISTATS*. PMLR. 2020, pp. 4519–4529 (cit. on p. 171).

[Kon+16] Jakub Konečnỳ et al. "Federated optimization: Distributed machine learning for on-device intelligence". In: *arXiv preprint arXiv:1610.02527* (2016) (cit. on pp. 3, 65, 75–77).

[Kon+21] Raed Kontar et al. "The Internet of Federated Things (IoFT): A Vision for the Future and In-depth Survey of Data-driven Approaches for Federated Learning". In: *arXiv preprint arXiv:2111.05326* (2021) (cit. on p. 80).

[Kos+15] Aris Kosmopoulos et al. "Evaluation measures for hierarchical classification: a unified view and novel approaches". In: *Data Mining and Knowledge Discovery* 29.3 (2015), pp. 820–865 (cit. on p. 134).

[KPK10] M Kumar, Benjamin Packer, and Daphne Koller. "Self-paced learning for latent variable models". In: *Advances in neural information processing systems* 23 (2010) (cit. on pp. 83, 89).

[KRM18] Md Abdullah Al Hafiz Khan, Nirmalya Roy, and Archan Misra. "Scaling human activity recognition via deep learning-based domain adaptation". In: *2018 IEEE international conference on pervasive computing and communications (PerCom)*. IEEE. 2018, pp. 1–9 (cit. on pp. 3, 8, 68–70).

[KS21] Mohammad Emtiyaz E Khan and Siddharth Swaroop. "Knowledge-Adaptation Priors". In: *Advances in Neural Information Processing Systems* 34 (2021) (cit. on p. 92).

[KSS19] Louis Kirsch, Sjoerd van Steenkiste, and Juergen Schmidhuber. "Improving Generalization in Meta Reinforcement Learning using Learned Objectives". In: *International Conference on Learning Representations*. 2019 (cit. on pp. 21, 35, 42).

[KT08] Charles Kemp and Joshua B Tenenbaum. "The discovery of structural form". In: *Proceedings of the National Academy of Sciences* 105.31 (2008), pp. 10687–10692 (cit. on p. 165).

[Kum+21] Sreejan Kumar et al. "Meta-Learning of Structured Task Distributions in Humans and Machines". In: *International Conference on Learning Representations*. 2021. URL: https://openreview.net/forum?id=--gvHfE3Xf5 (cit. on pp. 61, 136).

[KW13] Diederik P Kingma and Max Welling. "Auto-encoding variational bayes". In: *arXiv preprint arXiv:1312.6114* (2013) (cit. on p. 173).

[Lai84] Keith J Laidler. "The development of the Arrhenius equation". In: *Journal of Chemical Education* 61.6 (1984), p. 494 (cit. on p. 109).

[Lak+17] Brenden M Lake et al. "Building machines that learn and think like people". In: *Behavioral and brain sciences* 40 (2017) (cit. on p. 35).

[LB18] Thomas Laurent and James Brecht. "Deep linear networks with arbitrary loss: All local minima are global". In: *International conference on machine learning*. PMLR. 2018, pp. 2902–2907 (cit. on p. 34).

[LBG15] Christiane Lemke, Marcin Budka, and Bogdan Gabrys. "Metalearning: a survey of trends and technologies". In: *Artificial intelligence review* 44.1 (2015), pp. 117–130 (cit. on p. 37).

[LC18] Yoonho Lee and Seungjin Choi. "Gradient-based meta-learning with learned layerwise metric and subspace". In: *International Conference on Machine Learning*. PMLR. 2018, pp. 2927–2936 (cit. on pp. 53, 54).

[LH15] Ilya Loshchilov and Frank Hutter. "Online batch selection for faster training of neural networks". In: *arXiv preprint arXiv:1511.06343* (2015) (cit. on p. 89).

[Li+18] Hao Li et al. "Visualizing the loss landscape of neural nets". In: *Advances in neural information processing systems* 31 (2018) (cit. on pp. 127, 128).

[Li+19a] Xiang Li et al. "On the Convergence of FedAvg on Non-IID Data". In: *International Conference on Learning Representations*. 2019 (cit. on p. 75).

[Li+19b] Xingguo Li et al. "Symmetry, saddle points, and global optimization landscape of nonconvex matrix factorization". In: *IEEE Transactions on Information Theory* 65.6 (2019), pp. 3489–3514 (cit. on p. 205).

[Li+20a] Tian Li et al. "Federated optimization in heterogeneous networks". In: *MLSys* 2 (2020), pp. 429–450 (cit. on pp. 77, 171).

[Li+20b] Yonggang Li et al. “Dada: Differentiable automatic data augmentation”. In: *arXiv preprint arXiv:2003.03780* (2020) (cit. on pp. 103, 104).

[Lin+18] Yujun Lin et al. “Deep Gradient Compression: Reducing the Communication Bandwidth for Distributed Training”. In: *International Conference on Learning Representations*. 2018 (cit. on p. 3).

[Lin+21] Ming Lin et al. “Zen-nas: A zero-shot nas for high-performance deep image recognition”. In: *arXiv preprint arXiv:2102.01063* (2021) (cit. on p. 51).

[Lip18] Zachary C Lipton. “The Mythos of Model Interpretability: In machine learning, the concept of interpretability is both important and slippery.” In: *Queue* 16.3 (2018), pp. 31–57 (cit. on pp. 2, 56).

[Liu+17] Yuanyuan Liu et al. “Accelerated first-order methods for geodesically convex optimization on Riemannian manifolds”. In: *Advances in Neural Information Processing Systems* 30 (2017) (cit. on p. 206).

[LJ09] Mantas Lukoševičius and Herbert Jaeger. “Reservoir computing approaches to recurrent neural network training”. In: *Computer Science Review* 3.3 (2009), pp. 127–149 (cit. on p. 149).

[LL93] Chulhee Lee and David A Landgrebe. “Decision boundary feature extraction for nonparametric classification”. In: *IEEE transactions on systems, man, and cybernetics* 23.2 (1993), pp. 433–444 (cit. on p. 94).

[LM19] Mario Lezcano-Casado and David Martınez-Rubio. “Cheap orthogonal constraints in neural networks: A simple parametrization of the orthogonal and unitary group”. In: *International Conference on Machine Learning*. PMLR. 2019, pp. 3794–3803 (cit. on p. 206).

[Lon+19] Tatjana Loncar-Turukalo et al. “Literature on wearable technology for connected health: scoping review of research trends, advances, and barriers”. In: *Journal of medical Internet research* 21.9 (2019), e14017 (cit. on p. 2).

[Lop+15] David Lopez-Paz et al. “Unifying distillation and privileged information”. In: *arXiv preprint arXiv:1511.03643* (2015) (cit. on p. 59).

[LR17] David Lopez-Paz and Marc’Aurelio Ranzato. “Gradient episodic memory for continual learning”. In: *Advances in neural information processing systems* 30 (2017) (cit. on pp. 84, 85, 89).

[LR95] H Li and JF Reynolds. “On definition and quantification of heterogeneity”. In: *Oikos* (1995), pp. 280–284 (cit. on p. 67).

[Luc+20] James Lucas et al. "Theoretical bounds on estimation error for meta-learning". In: *International Conference on Learning Representations*. 2020 (cit. on p. 62).

[Luo+21] Mi Luo et al. "No fear of heterogeneity: Classifier calibration for federated learning with non-iid data". In: *Advances in Neural Information Processing Systems* 34 (2021) (cit. on pp. 72, 73).

[LW67] Godfrey N Lance and William Thomas Williams. "A general theory of classificatory sorting strategies: 1. Hierarchical systems". In: *The computer journal* 9.4 (1967), pp. 373–380 (cit. on p. 154).

[LYZ20] Sen Lin, Guang Yang, and Junshan Zhang. "A collaborative learning framework via federated meta-learning". In: *2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS)*. IEEE. 2020, pp. 289–299 (cit. on p. 80).

[Ma+19] Haojie Ma et al. "AttnSense: Multi-level Attention Mechanism For Multimodal Human Activity Recognition." In: *IJCAI*. 2019, pp. 3109–3115 (cit. on pp. 97, 145).

[Mad+14] Will Maddern et al. "Illumination invariant imaging: Applications in robust vision-based localisation, mapping and classification for autonomous vehicles". In: *Proceedings of the Visual Place Recognition in Changing Environments Workshop, IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China*. Vol. 2. 2014, p. 3 (cit. on pp. 102, 103).

[Mad+20] Spandan Madan et al. "When and how do CNNs generalize to out-of-distribution category-viewpoint combinations?" In: *arXiv preprint arXiv:2007.08032* (2020) (cit. on p. 171).

[MAL12] Samaneh Movassaghi, Mehran Abolhasan, and Justin Lipman. "Energy efficient thermal and power aware (ETPA) routing in body area networks". In: *2012 IEEE 23rd International Symposium on Personal, Indoor and Mobile Radio Communications-(PIMRC)*. IEEE. 2012, pp. 1108–1113 (cit. on p. 9).

[Mat+04] Merryn J Mathie et al. "Classification of basic daily movements using a triaxial accelerometer". In: *Medical and Biological Engineering and Computing* 42.5 (2004), pp. 679–687 (cit. on p. 15).

[Mat+19] Emile Mathieu et al. "Disentangling disentanglement in variational autoencoders". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 4402–4412 (cit. on pp. 176, 190).

[Mau+06] Uwe Maurer et al. "Activity recognition and monitoring using multiple sensors on different body positions". In: *International Workshop on Wearable and Implantable Body Sensor Networks (BSN'06)*. IEEE. 2006, 4–pp (cit. on pp. 13, 14).

[MC13] Aaron Mavrinac and Xiang Chen. "Modeling coverage in camera networks: A survey". In: *International journal of computer vision* 101.1 (2013), pp. 205–226 (cit. on p. 16).

[McM+17] Brendan McMahan et al. "Communication-efficient learning of deep networks from decentralized data". In: *Artificial Intelligence and Statistics*. PMLR. 2017, pp. 1273–1282 (cit. on pp. 3, 65, 66, 70, 194).

[Mel+21] Joe Mellor et al. "Neural architecture search without training". In: *International Conference on Machine Learning*. PMLR. 2021, pp. 7588–7598 (cit. on p. 51).

[MFF16] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. "Deepfool: a simple and accurate method to fool deep neural networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 2574–2582 (cit. on pp. 91, 93).

[MH08] Laurens van der Maaten and Geoffrey Hinton. "Visualizing data using t-SNE". In: *Journal of Machine Learning Research* 9.Nov (2008), pp. 2579–2605 (cit. on p. 100).

[Mis+18] Konstantin Mishchenko et al. "A delay-tolerant proximal-gradient algorithm for distributed learning". In: *International Conference on Machine Learning*. PMLR. 2018, pp. 3587–3595 (cit. on p. 3).

[Mit+18] Tom Mitchell et al. "Never-ending learning". In: *Communications of the ACM* 61.5 (2018), pp. 103–115 (cit. on p. 23).

[Mit+97] Tom M Mitchell et al. "Machine learning. 1997". In: *Burr Ridge, IL: McGraw Hill* 45.37 (1997), pp. 870–877 (cit. on p. 31).

[Mor07] Edgar Morin. "Restricted complexity, general complexity". In: *Science and us: Philosophy and Complexity. Singapore: World Scientific* (2007), pp. 1–25 (cit. on p. 4).

[Mor15] Edgar Morin. *Introduction à la pensée complexe*. Le Seuil, 2015 (cit. on p. 4).

[MPR16] Andreas Maurer, Massimiliano Pontil, and Bernardino Romera-Paredes. "The benefit of multitask representation learning". In: *Journal of Machine Learning Research* 17.81 (2016), pp. 1–32 (cit. on pp. 36, 62).

[MR17] Brendan McMahan and Daniel Ramage. *Collaborative machine learning without centralized training data*. https://ai.googleblog.com/2017/04/federated-learning-collaborative.html. Accessed: 2022-06-21. 2017 (cit. on pp. 3, 65).

[MSB20] Alejandro Melendez-Calderon, Camila Shirota, and Sivakumar Balasubramanian. "Estimating Movement Smoothness from Inertial Measurement Units". In: *bioRxiv* (2020) (cit. on pp. 168, 173).

[Nag+19] Anusha Nagabandi et al. "Learning to Adapt in Dynamic, Real-World Environments through Meta-Reinforcement Learning". In: *International Conference on Learning Representations*. 2019 (cit. on p. 87).

[Nag+20] Anusha Nagabandi et al. "Deep dynamics models for learning dexterous manipulation". In: *Conference on Robot Learning*. PMLR. 2020, pp. 1101–1112 (cit. on p. 60).

[NDC21] Cuong C Nguyen, Thanh-Toan Do, and Gustavo Carneiro. "Probabilistic task modelling for meta-learning". In: *Uncertainty in Artificial Intelligence*. PMLR. 2021, pp. 781–791 (cit. on pp. 61, 63, 136, 137).

[Ng11] Andrew Ng. "Sparse autoencoder". In: *CS294A Lecture notes* 72.2011 (2011), pp. 1–19 (cit. on p. 98).

[NS13] Tan Nguyen and Scott Sanner. "Algorithms for direct 0–1 loss optimization in binary classification". In: *International Conference on Machine Learning*. PMLR. 2013, pp. 1085–1093 (cit. on p. 33).

[Nun20] Abraham Nunes. "Measurement of Heterogeneity in Computational Psychiatry". PhD thesis. Dalhousie University, 2020 (cit. on p. 67).

[OBT21] Elre Talea Oldewage, John F Bronskill, and Richard E Turner. "Attacking Few-Shot Classifiers with Adversarial Support Poisoning". In: *ICML 2021 Workshop on Adversarial Machine Learning*. 2021 (cit. on p. 58).

[OH18] Aomar Osmani and Massinissa Hamidi. "Hybrid and convolutional neural networks for locomotion recognition". In: *Proceedings of the 2018 ACM UbiComp/ISWC 2018 Adjunct, Singapore, October 08-12, 2018*. ACM. 2018, pp. 1531–1540 (cit. on p. 29).

[OH19] Aomar Osmani and Massinissa Hamidi. "Bayesian optimization of neural architectures for human activity recognition". In: *Human Activity Sensing*. Springer, 2019, pp. 171–195 (cit. on p. 28).

[OH22] Aomar Osmani and Massinissa Hamidi. "Reduction of the Position Bias via Multi-level Learning for Activity Recognition". In: *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer. 2022, pp. 289–302 (cit. on pp. 26, 29, 76, 167).

[OHA21a] Aomar Osmani, Massinissa Hamidi, and Pegah Alizadeh. "Hierarchical Learning of Dependent Concepts for Human Activity Recognition". In: *PAKDD*. Springer. 2021 (cit. on pp. 26, 29, 131, 133).

[OHA21b] Aomar Osmani, Massinissa Hamidi, and Pegah Alizadeh. "Hierarchical Learning of Dependent Concepts for Human Activity Recognition". In: *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer. 2021 (cit. on p. 147).

[OHA22] Aomar Osmani, Massinissa Hamidi, and Pegah Alizadeh. "Clustering approach to solve hierarchical classification problem complexity". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 36. 7. 2022, pp. 7904–7912 (cit. on pp. 26, 28, 131, 133).

[OHB19] Aomar Osmani, Massinissa Hamidi, and Salah Bouhouche. "Monitoring of a Dynamical System Based on Autoencoders". In: *proceedings of the 28th International Joint Conference on Artificial Intelligence, IJCAI*. 2019, pp. 1836–1843 (cit. on pp. 25, 29, 83, 94).

[OHB21] Aomar Osmani, Massinissa Hamidi, and Salah Bouhouche. "Augmented Experiment in Material Engineering Using Machine Learning". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 10. 2021, pp. 9251–9258 (cit. on pp. 25, 29, 83, 85).

[OHC17a] Aomar Osmani, Massinissa Hamidi, and Abdelghani Chibani. "Machine Learning Approach for Infant Cry Interpretation". In: *Tools with Artificial Intelligence (ICTAI), 2017 IEEE 29th International Conference on*. IEEE. 2017, pp. 182–186 (cit. on pp. 29, 42).

[OHC17b] Aomar Osmani, Massinissa Hamidi, and Abdelghani Chibani. "Platform for Assessment and Monitoring of Infant Comfort". In: *2017 AAAI Fall Symposia, Arlington, Virginia, USA, November 9-11, 2017*. 2017, pp. 36–44 (cit. on pp. 27, 29, 42).

[OM13] Christian Henry Wijaya Oey and Sangman Moh. "A survey on temperature-aware routing protocols in wireless body sensor networks". In: *Sensors* 13.8 (2013), pp. 9860–9877 (cit. on pp. 14, 16, 17).

[OR16] Francisco Javier Ordóñez and Daniel Roggen. "Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition". In: *Sensors* 16.1 (2016), p. 115 (cit. on pp. 96, 97, 145, 171).

[Par+06] Juha Parkka et al. “Activity classification using realistic data from wearable sensors”. In: *IEEE Transactions on information technology in biomedicine* 10.1 (2006), pp. 119–128 (cit. on p. 15).

[PC22] Sayak Paul and Pin-Yu Chen. “Vision transformers are robust learners”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 36. 2. 2022, pp. 2071–2081 (cit. on p. 91).

[Per+17] Gabriel Pereyra et al. “Regularizing neural networks by penalizing confident output distributions”. In: *arXiv preprint arXiv:1701.06548* (2017) (cit. on p. 129).

[PL15] Anastasia Pentina and Christoph H Lampert. “Lifelong learning with non-iid tasks”. In: *Advances in Neural Information Processing Systems* 28 (2015) (cit. on p. 60).

[PRS19] Matthew E Peters, Sebastian Ruder, and Noah A Smith. “To tune or not to tune? adapting pretrained representations to diverse tasks”. In: *arXiv preprint arXiv:1903.05987* (2019) (cit. on pp. 133, 150, 153).

[QBC20] Robin Quessard, Thomas Barrett, and William Clements. “Learning Disentangled Representations and Group Structure of Dynamical Environments”. In: *Advances in Neural Information Processing Systems* 33 (2020) (cit. on pp. 170, 189).

[Qia+21] Hangwei Qian et al. “Latent Independent Excitation for Generalizable Sensor-based Cross-Person Activity Recognition”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 13. 2021, pp. 11921–11929 (cit. on pp. 175, 191).

[Qia99] Ning Qian. “On the momentum term in gradient descent learning algorithms”. In: *Neural networks* 12.1 (1999), pp. 145–151 (cit. on p. 75).

[Qui86] J. Ross Quinlan. “Induction of decision trees”. In: *Machine learning* 1.1 (1986), pp. 81–106 (cit. on p. 134).

[Rag+19] Aniruddh Raghu et al. “Rapid Learning or Feature Reuse? Towards Understanding the Effectiveness of MAML”. In: *International Conference on Learning Representations*. 2019 (cit. on pp. 50, 51, 53).

[Raj+19] Aravind Rajeswaran et al. “Meta-learning with implicit gradients”. In: *Advances in neural information processing systems* 32 (2019) (cit. on pp. 47, 86).

[Rau+17] Tifenn Rault et al. "A survey of energy-efficient context recognition systems using wearable sensors for healthcare applications". In: *Pervasive and Mobile Computing* 37 (2017), pp. 23–44 (cit. on pp. 11, 13).

[Rav+19a] Sathya N Ravi et al. "Explicitly imposing constraints in deep networks via conditional gradients gives improved generalization and faster convergence". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 2019, pp. 4772–4779 (cit. on p. 116).

[Rav+19b] Alice Ravizza et al. "Comprehensive review on current and future regulatory requirements on wearable sensors in preclinical and clinical testing". In: *Frontiers in bioengineering and biotechnology* 7 (2019), p. 313 (cit. on p. 16).

[Red+20] Sashank J Reddi et al. "Adaptive Federated Optimization". In: *International Conference on Learning Representations*. 2020 (cit. on pp. 74, 75).

[Ren+18] Mengye Ren et al. "Learning to reweight examples for robust deep learning". In: *International conference on machine learning*. PMLR. 2018, pp. 4334–4343 (cit. on pp. 83, 89).

[Ren+20] Jinke Ren et al. "Scheduling for cellular federated edge learning with importance and channel awareness". In: *IEEE Transactions on Wireless Communications* 19.11 (2020), pp. 7690–7703 (cit. on pp. 3, 11).

[Rif+11] Salah Rifai et al. "Contractive auto-encoders: Explicit invariance during feature extraction". In: *Proceedings of the 28th International Conference on Machine Learning*. Omnipress. 2011, pp. 833–840 (cit. on p. 98).

[Rit+07] Frank E Ritter et al. *In order to learn: How the sequence of topics influences learning*. Oxford University Press, 2007 (cit. on p. 126).

[RL16] Sachin Ravi and Hugo Larochelle. "Optimization as a model for few-shot learning". In: (2016) (cit. on p. 39).

[RSP17] Siamak Ravanbakhsh, Jeff Schneider, and Barnabas Poczos. "Equivariance through parameter-sharing". In: *International Conference on Machine Learning*. PMLR. 2017, pp. 2892–2901 (cit. on pp. 76, 165, 205).

[RST87] Larry A Rendell, Raj Sheshu, and David K Tcheng. "Layered Concept-Learning and Dynamically Variable Bias Management." In: *IJCAI*. Irvine, CA. 1987, pp. 308–314 (cit. on p. 35).

[RW12] Wolfgang Ring and Benedikt Wirth. "Optimization methods on Riemannian manifolds and their application to shape space". In: *SIAM Journal on Optimization* 22.2 (2012), pp. 596–627 (cit. on p. 206).

[Sad+20] Mostafa Sadeghi et al. "Audio-visual speech enhancement using conditional variational auto-encoders". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 28 (2020), pp. 1788–1800 (cit. on p. 175).

[Sal+16] S Ahmad Salehi et al. "IEEE 802.15. 6 standard in wireless body area networks from a healthcare point of view". In: *2016 22nd Asia-Pacific Conference on Communications (APCC)*. IEEE. 2016, pp. 523–528 (cit. on p. 16).

[SBH20] Farzad Samie, Lars Bauer, and Jörg Henkel. "Hierarchical Classification for Constrained IoT Devices: A Case Study on Human Activity Recognition". In: *IEEE Internet of Things Journal* (2020) (cit. on p. 134).

[Sch+20] Sebastian Scheurer et al. "Using domain knowledge for interpretable and competitive multi-class human activity recognition". In: *Sensors* 20.4 (2020), p. 1208 (cit. on p. 134).

[Sch16] Jürgen Schmidhuber. *Learning how to Learn Learning Algorithms: Recursive Self-Improvement*. 2016 (cit. on pp. 36–38).

[Sch87] Jürgen Schmidhuber. "Evolutionary principles in self-referential learning, or on learning how to learn: the meta-meta-... hook". PhD thesis. Technische Universität München, 1987 (cit. on pp. 4, 20, 22, 44, 85).

[Sch90] Robert E Schapire. "The strength of weak learnability". In: *Machine learning* 5.2 (1990), pp. 197–227 (cit. on p. 38).

[SE17] Russell Stewart and Stefano Ermon. "Label-free supervision of neural networks with physics and domain knowledge." In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 1. 1. 2017, pp. 2576–2582 (cit. on p. 113).

[Sev+19] Kristen A Severson et al. "Data-driven prediction of battery cycle life before capacity degradation". In: *Nature Energy* 4.5 (2019), p. 383 (cit. on p. 27).

[SF11] Carlos N Silla and Alex A Freitas. "A survey of hierarchical classification across different application domains". In: *Data Mining and Knowledge Discovery* 22.1-2 (2011), pp. 31–72 (cit. on pp. 132, 134, 143).

[Shi+20] Junhao Shi et al. "Sensor-based activity recognition independent of device placement and orientation". In: *Transactions on Emerging Telecommunications Technologies* 31.4 (2020), e3823 (cit. on pp. 168, 171).

[Sho+14] Muhammad Shoaib et al. "Fusion of smartphone motion sensors for physical activity recognition". In: *Sensors* 14.6 (2014), pp. 10146–10176 (cit. on pp. 179, 191).

[SK10] Sujesha Sudevalayam and Purushottam Kulkarni. "Energy harvesting sensor nodes: Survey and implications". In: *IEEE Communications Surveys & Tutorials* 13.3 (2010), pp. 443–461 (cit. on p. 9).

[SK19] Connor Shorten and Taghi M Khoshgoftaar. "A survey on image data augmentation for deep learning". In: *Journal of big data* 6.1 (2019), pp. 1–48 (cit. on p. 104).

[SKM19] Hiroyuki Sato, Hiroyuki Kasai, and Bamdev Mishra. "Riemannian stochastic variance reduced gradient algorithm with retraction and vector transport". In: *SIAM Journal on Optimization* 29.2 (2019), pp. 1444–1472 (cit. on p. 206).

[SLA12] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. "Practical bayesian optimization of machine learning algorithms". In: *NIPS*. 2012, pp. 2951–2959 (cit. on pp. 119, 145, 158).

[SLS21] Amrith Setlur, Oscar Li, and Virginia Smith. "Two Sides of Meta-Learning Evaluation: In vs. Out of Distribution". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 3770–3783 (cit. on p. 60).

[SMG13] Andrew M Saxe, James L McClelland, and Surya Ganguli. "Exact solutions to the nonlinear dynamics of learning in deep linear neural networks". In: *arXiv preprint arXiv:1312.6120* (2013) (cit. on p. 34).

[Smi+17] Virginia Smith et al. "Federated multi-task learning". In: *Advances in neural information processing systems* 30 (2017) (cit. on pp. 63, 79, 80, 86).

[SS10] T. Schaul and J. Schmidhuber. "Metalearning". In: *Scholarpedia* 5.6 (2010). revision #91489, p. 4650. DOI: 10.4249/scholarpedia.4650 (cit. on pp. 37, 38).

[SSZ17] Jake Snell, Kevin Swersky, and Richard Zemel. "Prototypical networks for few-shot learning". In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. 2017, pp. 4080–4090 (cit. on pp. 20, 35, 43).

[Sta+20] Trevor Standley et al. "Which tasks should be learned together in multi-task learning?" In: *International Conference on Machine Learning*. PMLR. 2020, pp. 9120–9132 (cit. on p. 136).

[Sti+15] Allan Stisen et al. "Smart devices are different: Assessing and mitigatingmobile sensing heterogeneities for activity recognition". In: *Proceedings of the 13th ACM conference on embedded networked sensor systems*. 2015, pp. 127–140 (cit. on pp. 3, 7, 8, 69–71, 171, 178).

[Sun+18] Flood Sung et al. "Learning to compare: Relation network for few-shot learning". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 1199–1208 (cit. on pp. 20, 35, 43).

[Sze+15] Christian Szegedy et al. "Going deeper with convolutions". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 1–9 (cit. on p. 34).

[Sze+16] Christian Szegedy et al. "Rethinking the inception architecture for computer vision". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 2818–2826 (cit. on p. 129).

[Tab+18] Daniel P Tabor et al. "Accelerating the discovery of materials for clean energy in the era of smart automation". In: *Nature Reviews Materials* 3.5 (2018), p. 5 (cit. on p. 27).

[Ten+11] Joshua B Tenenbaum et al. "How to grow a mind: Statistics, structure, and abstraction". In: *science* 331.6022 (2011), pp. 1279–1285 (cit. on p. 165).

[TJJ20] Nilesh Tripuraneni, Michael Jordan, and Chi Jin. "On the theory of transfer learning: The importance of task diversity". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 7852–7862 (cit. on p. 62).

[Tla+12] A Tlaisi et al. "Crack detection in shaft using lateral and torsional vibration measurements and analyses". In: *Turbo Expo: Power for Land, Sea, and Air*. Vol. 44731. American Society of Mechanical Engineers. 2012, pp. 693–702 (cit. on p. 100).

[TMF07] Antonio Torralba, Kevin P Murphy, and William T Freeman. "Sharing visual features for multiclass and multiview object detection". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29.5 (2007), pp. 854–869 (cit. on pp. 149, 164).

[TNH19] Anh T Tran, Cuong V Nguyen, and Tal Hassner. "Transferability and hardness of supervised classification tasks". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 1395–1405 (cit. on pp. 61, 136).

[TO20] Molei Tao and Tomoki Ohsawa. "Variational optimization on lie groups, with examples of leading (generalized) eigenvalue problems". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2020, pp. 4269–4280 (cit. on p. 206).

[Tob+12] Diego Alejandro Tobon-Mejia et al. "A data-driven failure prognostics method based on mixture of Gaussians hidden Markov models". In: *IEEE Transactions on reliability* 61.2 (2012), pp. 491–503 (cit. on p. 27).

[TP98] Sebastian Thrun and Lorien Pratt. "Learning to learn: Introduction and overview". In: *Learning to learn*. Springer, 1998, pp. 3–17 (cit. on pp. 32, 35, 36, 44, 52, 53, 55, 85).

[Tru+13] Alexander A Trusov et al. "Silicon accelerometer with differential frequency modulation and continuous self-calibration". In: *2013 IEEE 26th International Conference on Micro Electro Mechanical Systems (MEMS)*. IEEE. 2013, pp. 29–32 (cit. on p. 7).

[TTN20] Canh T Dinh, Nguyen Tran, and Tuan Dung Nguyen. "Personalized federated learning with Moreau envelopes". In: *Advances in Neural Information Processing Systems* 33 (2020) (cit. on pp. 74, 77, 177).

[Utg86a] Paul E Utgoff. "Machine learning of inductive bias". In: (1986) (cit. on p. 33).

[Utg86b] Paul E Utgoff. "Shift of bias for inductive concept learning". In: *Machine learning: An artificial intelligence approach* 2 (1986), pp. 107–148 (cit. on p. 35).

[VAC14] Raviteja Vemulapalli, Felipe Arrate, and Rama Chellappa. "Human action recognition by representing 3d skeletons as points in a lie group". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pp. 588–595 (cit. on p. 188).

[Val84] Leslie G Valiant. "A theory of the learnable". In: *Communications of the ACM* 27.11 (1984), pp. 1134–1142 (cit. on pp. 19, 32).

[Van19] Joaquin Vanschoren. "Meta-learning". In: *Automated Machine Learning*. Springer, Cham, 2019, pp. 35–61 (cit. on pp. 59, 60).

[Vap91] Vladimir Vapnik. "Principles of risk minimization for learning theory". In: *Advances in neural information processing systems* 4 (1991) (cit. on p. 202).

[Vap92] Vladimir Vapnik. "Principles of risk minimization for learning theory". In: *Advances in neural information processing systems*. 1992, pp. 831–838 (cit. on p. 127).

[Vap95] Vladimir N. Vapnik. *The Nature of Statistical Learning Theory*. Berlin, Heidelberg: Springer-Verlag, 1995. ISBN: 0387945598 (cit. on pp. 48, 62).

[Vas+17] Ashish Vaswani et al. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017) (cit. on pp. 39, 205).

[VC82] Vladimir N Vapnik and A Ya Chervonenkis. "Necessary and sufficient conditions for the uniform convergence of means to their expectations". In: *Theory of Probability & Its Applications* 26.3 (1982), pp. 532–553 (cit. on pp. 19, 32).

[VD02] Ricardo Vilalta and Youssef Drissi. "A perspective view and survey of meta-learning". In: *Artificial intelligence review* 18.2 (2002), pp. 77–95 (cit. on pp. 21, 35, 49).

[VI15] Vladimir Vapnik and Rauf Izmailov. "Learning using privileged information: similarity control and knowledge transfer." In: *Journal of Machine Learning Research* 16.2023-2049 (2015), p. 2 (cit. on pp. 58, 84, 89).

[VI20] Vladimir Vapnik and Rauf Izmailov. "Complete statistical theory of learning: learning using statistical invariants". In: *Conformal and Probabilistic Prediction and Applications*. PMLR. 2020, pp. 4–40 (cit. on pp. 84, 170).

[Vin+10] Pascal Vincent et al. "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion." In: *Journal of machine learning research* 11.12 (2010) (cit. on p. 152).

[Vin+16] Oriol Vinyals et al. "Matching networks for one shot learning". In: *Advances in neural information processing systems* 29 (2016), pp. 3630–3638 (cit. on pp. 21, 35, 43).

[Vuo+19] Risto Vuorio et al. "Multimodal model-agnostic meta-learning via task-aware modulation". In: *Advances in Neural Information Processing Systems* 32 (2019) (cit. on pp. 63, 64).

[VV09] Vladimir Vapnik and Akshay Vashist. "A new learning paradigm: Learning using privileged information". In: *Neural networks* 22.5-6 (2009), pp. 544–557 (cit. on pp. 84, 89).

[Wan+13] Ning Wang et al. "Energy and accuracy trade-offs in accelerometry-based activity recognition". In: *2013 22nd International Conference on Computer Communication and Networks (ICCCN)*. IEEE. 2013, pp. 1–6 (cit. on p. 13).

[Wan+20a] Hongyi Wang et al. "Federated Learning with Matched Averaging". In: *International Conference on Learning Representations*. 2020 (cit. on pp. 76, 77, 204).

[Wan+20b] Jianyu Wang et al. "Tackling the Objective Inconsistency Problem in Heterogeneous Federated Optimization". In: *Advances in Neural Information Processing Systems* 33 (2020) (cit. on pp. 3, 9).

[Wan+21] Jianyu Wang et al. "A field guide to federated optimization". In: *arXiv preprint arXiv:2107.06917* (2021) (cit. on p. 32).

[Wan+22] Haoxiang Wang et al. "Global Convergence of MAML and Theory-Inspired Neural Architecture Search for Few-Shot Learning". In: *arXiv preprint arXiv:2203.09137* (2022) (cit. on p. 51).

[Wat+15] Manuel Watter et al. "Embed to control: a locally Linear Latent dynamics model for control from raw images". In: *Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 2*. 2015, pp. 2746–2754 (cit. on p. 197).

[WCA18] Daphna Weinshall, Gad Cohen, and Dan Amir. "Curriculum learning by transfer learning: Theory and experiments with deep networks". In: *International Conference on Machine Learning*. PMLR. 2018, pp. 5238–5246 (cit. on p. 89).

[WCB18] Jonatas Wehrmann, Ricardo Cerri, and Rodrigo Barros. "Hierarchical multi-label classification networks". In: *International Conference on Machine Learning*. 2018, pp. 5075–5084 (cit. on p. 132).

[WKA10] Chen Wu, Amir Hossein Khalili, and Hamid Aghajan. "Multiview activity recognition in smart homes with spatio-temporal features". In: *Proceedings of the fourth ACM/IEEE international conference on distributed smart cameras*. 2010, pp. 142–149 (cit. on pp. 15, 81, 173).

[WKS16] Jialei Wang, Mladen Kolar, and Nathan Srerbo. "Distributed multi-task learning". In: *Artificial intelligence and statistics*. PMLR. 2016, pp. 751–760 (cit. on pp. 79, 86).

[WZL21] Haoxiang Wang, Han Zhao, and Bo Li. "Bridging Multi-Task Learning and Meta-Learning: Towards Efficient Training and Effective Adaptation". In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, 18–24 Jul 2021, pp. 10991–11002 (cit. on p. 36).

[Xie+16] Lingxi Xie et al. "Disturblabel: Regularizing cnn on the loss layer". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 4753–4762 (cit. on p. 129).

[Xu+21a] Han Xu et al. "Yet meta learning can adapt fast, it can also break easily". In: *Proceedings of the 2021 SIAM International Conference on Data Mining (SDM)*. SIAM. 2021, pp. 540–548 (cit. on p. 58).

[Xu+21b] Jingjing Xu et al. "KNAS: green neural architecture search". In: *International Conference on Machine Learning*. PMLR. 2021, pp. 11613–11625 (cit. on p. 51).

[Yao+17] Shuochao Yao et al. "Deepsense: A unified deep learning framework for time-series mobile sensing data processing". In: *Proceedings of the 26th International Conference on World Wide Web*. 2017, pp. 351–360 (cit. on pp. 97, 145, 171).

[Yao+19] Huaxiu Yao et al. "Hierarchically Structured Meta-learning". In: *International Conference on Machine Learning*. 2019, pp. 7045–7054 (cit. on pp. 61, 132, 136).

[Yeo+21] De Jong Yeong et al. "Sensor and sensor fusion technology in autonomous vehicles: A review". In: *Sensors* 21.6 (2021), p. 2140 (cit. on pp. 3, 8).

[Yu+14] Meng-Chieh Yu et al. "Big data small footprint: the design of a low-power classifier for detecting transportation modes". In: *Proceedings of the VLDB Endowment* 7.13 (2014), pp. 1429–1440 (cit. on p. 145).

[Yu+18] Tianhe Yu et al. "One-shot imitation from observing humans via domain-adaptive meta-learning". In: *arXiv preprint arXiv:1802.01557* (2018) (cit. on p. 42).

[Yu+20] Tianhe Yu et al. "Gradient surgery for multi-task learning". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 5824–5836 (cit. on p. 55).

[YW16] Rong Yang and Baowei Wang. "PACP: a position-independent activity recognition method using smartphone sensors". In: *Information* 7.4 (2016), p. 72 (cit. on p. 172).

[YWC08] Jhun-Ying Yang, Jeen-Shing Wang, and Yen-Ping Chen. "Using acceleration measurements for activity recognition: An effective learning algorithm for constructing neural classifiers". In: *Pattern recognition letters* 29.16 (2008), pp. 2213–2220 (cit. on pp. 15, 173).

[YYZ19] Hao Yu, Sen Yang, and Shenghuo Zhu. "Parallel restarted sgd with faster convergence and less communication: Demystifying why model averaging works for deep learning". In: *AAAI*. Vol. 33. 01. 2019, pp. 5693–5700 (cit. on p. 171).

[Zah+18] Manzil Zaheer et al. "Adaptive methods for nonconvex optimization". In: *Advances in neural information processing systems* 31 (2018) (cit. on p. 75).

[Zam+18] Amir R Zamir et al. "Taskonomy: Disentangling task transfer learning". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 3712–3722 (cit. on pp. 61, 133, 136, 150, 153).

[ZCY11] Jiayu Zhou, Jianhui Chen, and Jieping Ye. "Clustered multi-task learning via alternating structure optimization". In: *Advances in neural information processing systems* 24 (2011) (cit. on p. 63).

[Zha+18] Yue Zhao et al. "Federated learning with non-iid data". In: *arXiv preprint arXiv:1806.00582* (2018) (cit. on pp. 72–74).

[Zha+19] Cheng Zhang et al. "Active mini-batch sampling using repulsive point processes". In: *Proceedings of the AAAI conference on Artificial Intelligence*. Vol. 33. 01. 2019, pp. 5741–5748 (cit. on p. 83).

[Zho+21a] Fengwei Zhou et al. "Metaaugment: Sample-aware data augmentation policy learning". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 12. 2021, pp. 11097–11105 (cit. on p. 105).

[Zho+21b] Pan Zhou et al. "Task similarity aware meta learning: Theory-inspired improvement on maml". In: *Uncertainty in Artificial Intelligence*. PMLR. 2021, pp. 23–33 (cit. on pp. 64, 136).

[Zin+19] Luisa Zintgraf et al. "Fast context adaptation via meta-learning". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 7693–7702 (cit. on pp. 53, 54).

[ZJS16] Hongyi Zhang, Sashank J Reddi, and Suvrit Sra. "Riemannian SVRG: Fast stochastic optimization on Riemannian manifolds". In: *Advances in Neural Information Processing Systems* 29 (2016) (cit. on p. 206).

[ZL16] Barret Zoph and Quoc V Le. "Neural architecture search with reinforcement learning". In: *arXiv preprint arXiv:1611.01578* (2016) (cit. on pp. 20, 35, 41, 42).

[ZQW20] Yuqian Zhang, Qing Qu, and John Wright. "From symmetry to geometry: Tractable nonconvex problems". In: *arXiv preprint arXiv:2007.06753* (2020) (cit. on pp. 128, 205).

[ZS12] Mi Zhang and Alexander A Sawchuk. "USC-HAD: a daily activity dataset for ubiquitous activity recognition using wearable sensors". In: *UbiComp*. 2012, pp. 1036–1043 (cit. on p. 145).

[ZXW11] Dengyong Zhou, Lin Xiao, and Mingrui Wu. "Hierarchical classification via orthogonal transfer". In: *Proceedings of the 28th International Conference on International Conference on Machine Learning*. 2011, pp. 801–808 (cit. on pp. 132, 149, 164).

[ZY12] Yu Zhang and Dit-Yan Yeung. "A convex formulation for learning task relationships in multi-task learning". In: *arXiv preprint arXiv:1203.3536* (2012) (cit. on p. 63).