Domain Generalization in Machine Learning Models for Wireless Communications: Concepts, State-of-the-Art, and Open Issues

Mohamed Akrout, Amal Feriani, Faouzi Bellili, *Member, IEEE*, Amine Mezghani, *Member, IEEE*, and Ekram Hossain, *Fellow, IEEE*

***Abstract*—Data-driven machine learning (ML) is promoted as one potential technology to be used in next-generation wireless systems. This led to a large body of research work that applies ML techniques to solve problems in different layers of the wireless transmission link. However, most of these applications rely on supervised learning which assumes that the source (training) and target (test) data are independent and identically distributed (i.i.d). This assumption is often violated in the real world due to domain or distribution shifts between the source and the target data. Thus, it is important to ensure that these algorithms generalize to out-of-distribution (OOD) data. In this context, domain generalization (DG) tackles the OOD-related issues by learning models on different and distinct source domains/datasets with generalization capabilities to unseen new domains without additional finetuning. Motivated by the importance of DG requirements for wireless applications, we present a comprehensive overview of the recent developments in DG and the different sources of domain shift. We also summarize the existing DG methods and review their applications in selected wireless communication problems, and conclude with insights and open questions.**

***Index Terms*—ML-aided wireless networks, out-of-distribution generalization, domain generalization.**

I. Introduction

A. Motivation

THE ENVISIONED design, standardization,[1] and deployment of ML in wireless networks requires the establishment of evaluation guidelines to properly assess the true potential of data-driven methods. Nevertheless, almost all the openly published ML techniques for wireless systems have several limitations such as *i)* difficulty to generalize under a *distribution shift*, *ii)* inability to continuously learn from different scenarios, and *iii)* inability to *quickly* adapt to unseen scenarios, to name a few. Their showcased performance seems over-fitted to a specific set of simulation settings or fixed datasets, thereby limiting their attractiveness to compete with classical methods at the moment. As one example, the linear minimum mean-square error (LMMSE) estimator of an arbitrary channel model is considered by the industry as one of the most robust estimators in practice. While it is always possible to beat the LMMSE estimator with deep neural network (DNNs) approximators [2], this fact holds only for an *a priori* known model that is used to generate training and test datasets on which DNNs are trained and also evaluated. When the distributions of training and test datasets are different (e.g., Ricean vs. Rayleigh, or sparse vs. rich-scattering channels), the performance of DNNs deteriorates appreciably due to domain distribution gaps. Furthermore, the lack of real-world wireless communication datasets aggravates the uncertainty toward the practical deployment of ML-based methods. This calls for the development of new ML training algorithms and the establishment of rigorous evaluation protocols to assess their OOD generalization.

In this work, we focus on generalization under domain shift. This includes any change in the distribution between the training (i.e., source) data and the target (i.e., test data).[2] The most studied type of distribution shift is *covariate* shift when the distribution of the model inputs (or features) changes between the source and the target domains [3]. It was shown that the performance of DNNs degrades drastically due to small variations or perturbations in the training datasets [4]. Thus, the acclaimed success of deep learning (DL) is mostly driven by the power of supervised learning. One straightforward idea to overcome domain shift is to adapt the model to the new domain via additional finetuning using techniques such as transfer learning [5] and domain adaptation [6]. However, this is not always feasible in practice because *i)* target *labeled* data may not be available for finetuning and *ii)* the finetuning or adaptation may take a long time in contrast to the "real time" requirement in most wireless applications. This motivates the DG problem [7] to handle domain shift *without* requiring target domain data.

DG has been extensively studied in the last decade in the ML community which led to a broad spectrum of methodologies and learning techniques. Moreover, DG was examined in

Manuscript received 13 March 2023; revised 3 September 2023; accepted 18 October 2023. Date of publication 20 October 2023; date of current version 22 November 2023. This work was supported by the Discovery Grant from the Natural Sciences and Engineering Research Council of Canada (NSERC). *(Corresponding author: Ekram Hossain.)*

The authors are with the Department of Electrical and Computer Engineering, University of Manitoba, Winnipeg, MB R2M 2J8, Canada (e-mail: akroutm@myumanitoba.ca; feriania@myumanitoba.ca; faouzi.bellili@umanitoba.ca; amine.mezghani@umanitoba.ca; ekram.hossain@umanitoba.ca).

Digital Object Identifier 10.1109/COMST.2023.3326399

[1] See 3GPP Release 18 [1, Sec. IX.2] for some potential use cases of ML in wireless.

[2] The terms "training" (resp. "test") and "source" (resp. "target") are used interchangeably throughout the paper.

1553-877X © 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Northeastern University. Downloaded on February 26,2024 at 14:58:14 UTC from IEEE Xplore. Restrictions apply.

different applications, namely, computer vision [8], [9], natural language processing [10], [11], and medical imaging [12], etc. Here, we emphasize the importance of the DG problem in wireless applications to advance the current state-of-the-art research and raise attention to the problem of domain shift which can seriously impede the success of ML techniques in wireless networks. Specifically, we highlight the importance of leveraging wireless communication domain knowledge to tailor or design more generalizable ML algorithms.

This work provides a timely and comprehensive overview of the DG research landscape and insights into promising future research directions. The scope of this paper is limited to the DG problem as defined above. At the time of the writing, we have identified several DG variants proposed in the literature that we will briefly discuss but we focus on the standard definition of the DG problem. Other related fields such as domain adaptation, transfer learning, zero-shot learning, multi-task learning, and test time training are beyond the scope of this work. However, we will explain the difference between these fields and DG. In addition, we do distinguish between the terms "generalization" and "robustness", unlike most wireless communication papers which use them interchangeably. Here, generalization which is also known as *model robustness* denotes the ability of DNNs to generalize to unseen scenarios under distribution shifts. Robustness, however, refers to the stability of DNNs' performance under noise and adversarial examples, i.e., *adversarial robustness* [13].

B. Contributions and Organization of the Paper

The main contributions of this paper are summarized as follows:

- We define the DG problem and present four types of distribution shifts. We then contrast DG to existing research fields such as domain adaptation, transfer learning, continual learning, etc.
- We summarize different ML methodologies for DG which focus on the following three DNN training steps: *i)* data manipulation to cover richer domains pertaining to a given dataset, *ii)* representation learning to acquire domain-invariant features enabling generalization, and *iii)* learning frameworks which go beyond the standard gradient-based DNN optimization.
- We also review the literature on previous attempts to apply ML techniques for DG in several wireless communications problems such as channel decoding, beamforming, multiple-input multiple-output (MIMO) channel estimation, and reconfigurable intelligent surface (RIS)-aided communications. To the best of our knowledge, this is the first attempt to outline the existing applications of ML techniques in wireless research from the DG perspective.
- We present the main challenges facing the application of data-driven machine learning techniques in wireless communication under DG requirements and discuss their potential for improving the network performance.

The rest of the paper is organized as illustrated in Fig. 1. In Section II, we introduce the DG problem formulation and show its key differences with related research fields. State-of-the-art algorithms for DG belonging to data manipulation, representation learning, and learning paradigms are reviewed in Sections III-A–III-C, respectively. Section IV showcases the recent advances of the reviewed DG algorithms in several wireless communication problems, followed by a summary of the learned lessons from their applications. Finally, we outline in Section V potential research directions, from which we draw out our concluding remarks in Section VI.

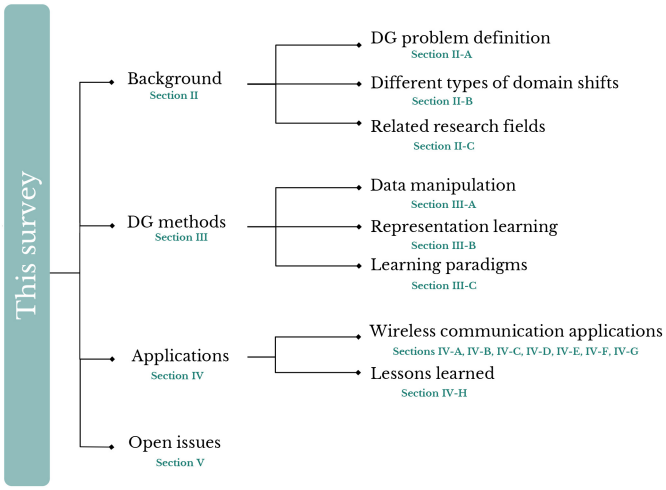


Fig. 1. Scope of this work.

We also mention the common notations used in this paper. We use Sans Serif fonts (e.g., x) for random variables and Serif fonts (e.g., x) for their realizations. We use boldface lowercase letters for vectors (e.g., x and $\boldsymbol{x}$). We also use $\mathbb{E}[\mathsf{x}]$ to denote the expectation of the random variable x.

II. Background

In this section, we will define the scope of the tutorial and highlight the resemblance between domain generalization and other related research fields. We start by introducing the following definitions.

A. DG Problem Formulation

Definition 1 (Domain): Let $\mathcal{X}$ and $\mathcal{Y}$ be the input space and the output space of a dataset $\mathcal{D} = \{(X_i, Y_i)|P_{\mathsf{X},\mathsf{Y}}(X_i, Y_i)\}_{i=1}^{K}$, where K is the size of the dataset, $X_i \in \mathcal{X}$ and $Y_i \in \mathcal{Y}$ are the i-th input and label samples, respectively. When X_i and Y_i are seen as realizations of their respective random variables X and Y, it is possible to define a domain as their joint distribution $P_{\mathsf{X},\mathsf{Y}}(X, Y)$. Moreover, $P_{\mathsf{X}}(X)$ and $P_{\mathsf{Y}}(Y)$ refer to the marginal distribution of X and Y, respectively. Throughout the paper, we will drop distribution arguments to lighten the notation.

Machine learning algorithms use one or multiple datasets, and as such make use of one or multiple data domains according to Definition 1. Indeed, training and evaluating ML techniques require *at least* two domains:

- a source domain $P^s_{\mathsf{X},\mathsf{Y}}$ encoding both the source input marginal P^s_{X} and the source label marginal P^s_{Y};

Authorized licensed use limited to: Northeastern University. Downloaded on February 26,2024 at 14:58:14 UTC from IEEE Xplore. Restrictions apply.

TABLE I
SUMMARY OF EXISTING SURVEYS ON DOMAIN GENERALIZATION

Paper	Summary	Target community
[7]	Present a literature review in DG summarizing the developments over the past decade	Machine learning
[14]	Provide a comprehensive review of the theory and recent advancements in DG	Machine learning
[15]	Provide an overview of DG approaches from a causal perspective	Machine learning
[16]	Investigate and implement different DG methods for image-based datasets	Machine learning
[17]	Present a benchmark of 10 datasets reflecting a diverse range of distribution shifts	Machine Learning
[18]	Introduce ten time-series benchmarks covering a diverse range of data modalities	Machine learning, IoT
Our Work	Outline the key DG methods and their applications in wireless communication	Wireless Networks

- a target domain $P^t_{\mathsf{X},\mathsf{Y}}$ encoding both the target input marginal P^t_{X} and the target label marginal P^t_{Y}.

Generalization is an active ML research area where the ultimate goal is to learn models that perform well on unseen data domains. This tutorial focuses on out-of-distribution generalization or domain generalization (DG). In the next section, we define the DG problem and explain the difference between this subfield and other generalization problems in ML.

Definition 2 (Domain Generalization): The traditional setting of DG consists of *M distinct* source datasets, i.e., $\mathcal{D}_{\text{train}} = \{\mathcal{D}^s\}_{s=1}^M$ with $\mathcal{D}^s = \{(X_i^s, Y_i^s, d_i^s)|P_{\mathsf{X}^s,\mathsf{Y}^s}(X_i^s, Y_i^s)\}_i$. Here, the ith data-target sample pair (X_i^s, Y_i^s) is sampled from the domain P^s_{XY} pertaining to the dataset $\mathcal{D}^s$, i.e., $(X_i^s, Y_i^s) \sim P^s_{\mathrm{XY}}$, and d_i^s is a label that is used to distinguish the key characteristics of the domain to which the data-target samples belong, e.g., radar, mmWave transmission, etc. DG also considers unseen target datasets $\mathcal{D}_{\text{test}} = \{\mathcal{D}^t\}_{t=1}^T$ which are different from the source datasets (i.e., $\mathcal{D}^t = \{(X_j^t, Y_j^t)|P_{\mathsf{X}^t,\mathsf{Y}^t}(X_i^t, Y_i^t)\}_j \neq \mathcal{D}^s$, $\forall s$ for $1 \leq s \leq M$). The goal of DG is to train on the source domains a model f that generalizes to the target domains under two key assumptions. The first one is *the violation of the i.i.d assumption* because the source and target samples are drawn from the same distribution. The second one is that training is performed *without any access to the target data*. The generalization is often measured via a loss function $\mathcal{L}(\cdot,\cdot)$ on the test domains, i.e., $\mathbb{E}_{\mathsf{X}^t,\mathsf{Y}^t}[\mathcal{L}(f(X_j^t), Y_j^t)]$. Different variations of the vanilla DG described above have been studied in the literature:

- *Single-source DG* assumes that there is a single domain (i.e., $M = 1$) accessible during training. All the training data are sampled from the same domain. In this case, the model is trained on data from a single source domain and is required to generalize to unseen target domains;
- *Homogeneous DG* requires the source and target domains to share the same label space, i.e., $\mathcal{Y}^s = \mathcal{Y}^t$;
- *Heterogeneous DG* assumes different label spaces for the source and target domains, i.e., $\mathcal{Y}^s \neq \mathcal{Y}^t$;
- *Compound DG:* The vanilla DG setting assumes that source domain labels d^s are known prior to learning. In contrast, compound DG does not require domain annotations and assumes that the source data is *heterogeneous* and consists of mixed domains. In other words, the training is not divided into distinct domains before learning. Thus, in addition to generalizing to new unseen domains, compound DG methods need to infer/learn domain information from mixed heterogeneous datasets. For this reason, compound DG is more challenging than vanilla DG.

Fig. 2 illustrates the difference between the vanilla and compound DG settings for estimating a wireless communication channel. There, we consider the wireless multi-path (MP) channel model $\boldsymbol{H}^{\text{MP}}(L, f)$ parametrized by the number of paths L and the frequency f. In vanilla DG, channel samples within the source dataset belong to *known* domains, i.e., the number of paths L is known for each sample. This is illustrated in Fig. 2(a) by clustering channel samples into three a priori known domains pertaining to three MP channel models associated with $L = 1, 5, 15$. In compound DG, however, the domains of the channel samples are *not known* as highlighted in Fig. 2(b). Hence, the source dataset can be perceived as an unlabeled dataset where the domain knowledge of samples is unknown. In summary, prior knowledge of "which samples belong to which domain" is the main difference between vanilla DG different from compound DG.

Before delving into the details of DG algorithms, we now present the possible distribution shifts in the source/target domains and the related DG research fields.

B. Different Types of Domain Shifts

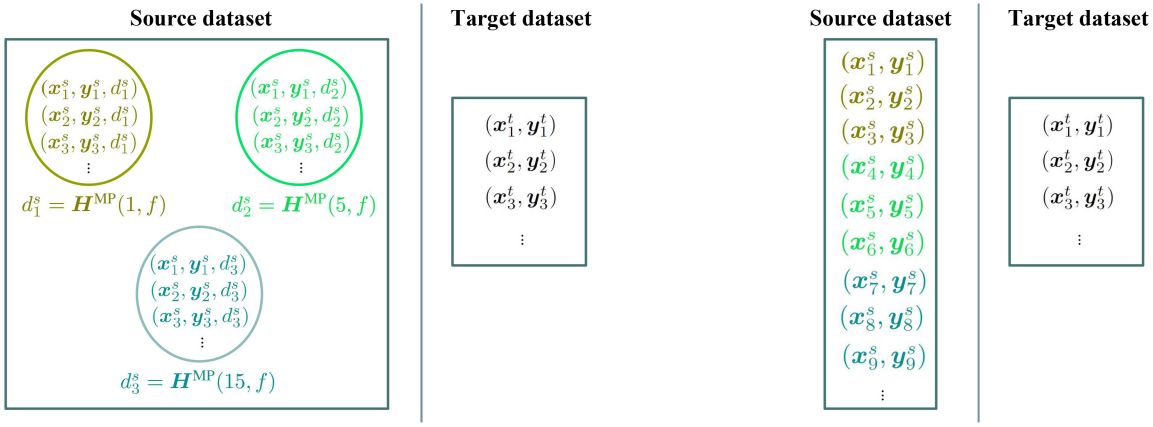
Given a joint distribution $P_{\mathsf{X},\mathsf{Y}}$ associated with either a source or target domain, it is always possible to factorize it in two different forms using the Bayes rule:

$$P_{\mathsf{X},\mathsf{Y}} = \underbrace{P_{\mathsf{X}|\mathsf{Y}}}_{\substack{\text{conditional}\\ \text{distribution}}} \times \underbrace{P_{\mathsf{Y}}}_{\substack{\text{label}\\ \text{distribution}}}, \tag{1a}$$

$$= \underbrace{P_{\mathsf{Y}|\mathsf{X}}}_{\substack{\text{concept}\\ \text{distribution}}} \times \underbrace{P_{\mathsf{X}}}_{\substack{\text{input}\\ \text{distribution}}}. \tag{1b}$$

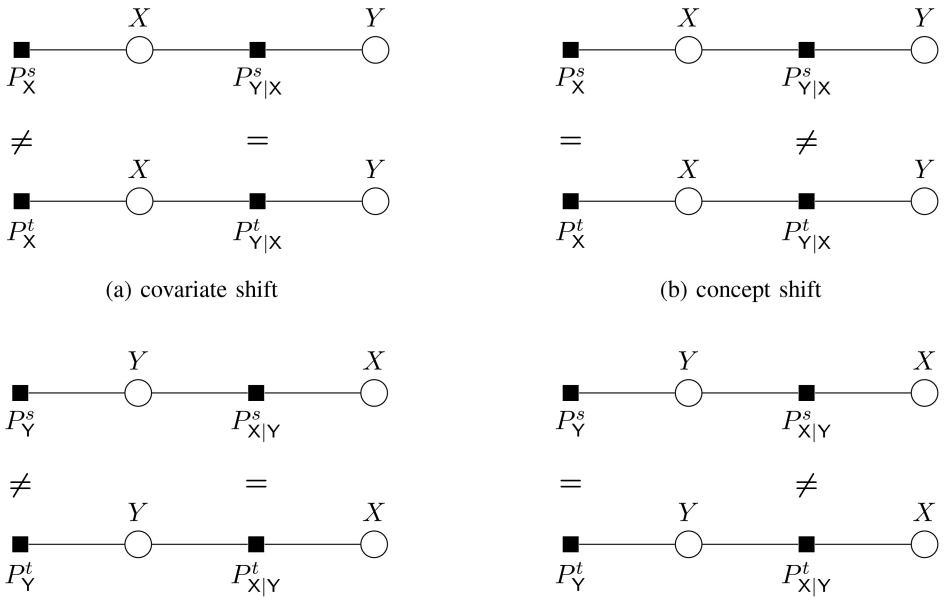
While the two domain factorizations in (1a) and (1b) are mathematically equivalent, they reveal two distinct directions of the causal relationship between the input random variable X and the label random variable Y. By way of explanation, the factorization in (1a) captures the causal relationship from Y to X. This is because the conditional distribution of X given Y cannot be determined unless a particular realization of Y has been already observed to fully characterize the conditional distribution $P_{\mathsf{X}|\mathsf{Y}}$. Exchanging the roles of Y and X shows how the causal relationship from X to Y is captured by the factorization in (1b). To see how this is the case, we resort in what follows to factor graphs as pictorial representations of joint probability distributions to efficiently communicate the

Authorized licensed use limited to: Northeastern University. Downloaded on February 26,2024 at 14:58:14 UTC from IEEE Xplore. Restrictions apply.



(a) Vanilla DG (b) Compound DG

Fig. 2. Two types of DG for a wireless multi-path channel estimation problem with data domains $d = \boldsymbol{H}^{\mathrm{MP}}(L, f)$.



(a) covariate shift (b) concept shift

(c) label shift (d) conditional shift

Fig. 3. Factor graphs of the four possible distribution shifts in the source and target domain factorizations $P^s_{\mathsf{X},\mathsf{Y}}$ and $P^t_{\mathsf{X},\mathsf{Y}}$: (a) covariate shift where $P^s_{\mathsf{X}} \neq P^t_{\mathsf{X}}$, (b) concept shift where $P^s_{\mathsf{Y}|\mathsf{X}} \neq P^t_{\mathsf{Y}|\mathsf{X}}$, (c) label shift where $P^s_{\mathsf{Y}} \neq P^t_{\mathsf{Y}}$, and (d) conditional shift where $P^s_{\mathsf{X}|\mathsf{Y}} \neq P^t_{\mathsf{X}|\mathsf{Y}}$.

conditional dependence structure between X and Y [19]. As depicted in Fig. 3, random variables correspond to "variables nodes" represented in circles while distributions are associated with "factor nodes" illustrated in squares. Each variable node is connected to a factor node through an edge only when the factor node is dependent on the variable node.

In general, distribution shifts can influence at least one of the four distributions involved in the domain factorization in (1). To better illustrate distribution shifts in wireless applications, let us consider a MIMO frequency-dependent communication setting where the received signal $\boldsymbol{y}$ of a baseband transmitted signal $\boldsymbol{x}$ over a MIMO channel $\boldsymbol{H}$ is given by:

$$\boldsymbol{y} = \boldsymbol{H}\,\boldsymbol{x} + \boldsymbol{w}, \tag{2}$$

where $\boldsymbol{w}$ is an additive white Gaussian noise (AWGN) vector whose entries are assumed to be mutually independent with mean zero and variance γ_w, i.e., $w_i \sim \mathcal{N}(\mathsf{w}_i; 0, \gamma_w^{-1})$.

We distinguish four types of distribution shifts between the source and target domains. Fig. 3 depicts the factor graphs associated with each of the following distribution shifts:

- a distribution shift between the source and target input distributions, i.e., $P^s_{\mathsf{X}} \neq P^t_{\mathsf{X}}$, as shown in Fig. 3(a). This shift is commonly called *covariate shift* [20] and is the most studied type of distribution shift in the literature. In (2), it is associated with a shift in the distribution of the pass-band signal. This can occur depending on the portion of the frequency spectrum in the signal that is transmitted after filtering.
- a distribution shift between the source and target concept distributions, i.e., $P^s_{\mathsf{Y}|\mathsf{X}} \neq P^t_{\mathsf{Y}|\mathsf{X}}$, as shown in Fig. 3(b). The concept shift is usually not examined in DG classification tasks because most of the prior work assumes that data samples have different labels in different domains. In (2), this shift is associated with the distribution

Authorized licensed use limited to: Northeastern University. Downloaded on February 26,2024 at 14:58:14 UTC from IEEE Xplore. Restrictions apply.

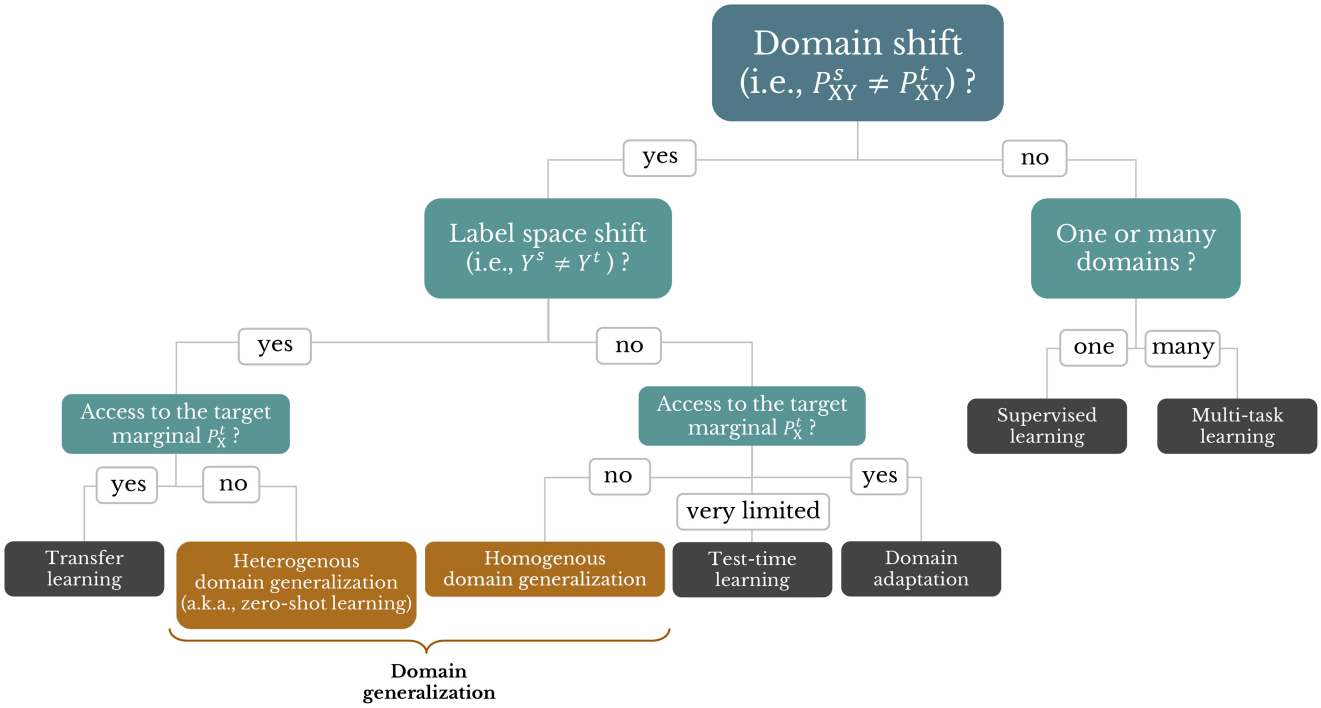


Fig. 4. Similarities and differences between domain generalization covered in this work and other generalization-related topics.

$p_{\mathsf{Y}|\mathsf{X}}(\boldsymbol{y}|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{H}\,\boldsymbol{x}, \gamma_w^{-1}\,\mathbf{I})$. Assuming no change in the distribution $p_{\mathsf{X}}(\boldsymbol{x})$ of the signal $\boldsymbol{x}$, this scenario to a change in the noise variance γ_w^{-1}, and hence the SNR.

- a distribution shift between the source and target label distributions, i.e., $P^s_{\mathsf{Y}} \neq P^t_{\mathsf{Y}}$, as illustrated in Fig. 3(c). This is called *label shift* and is common in ML datasets, e.g., class imbalance in classification tasks. In (2), this shift is associated with the distribution of the received signal which can be caused by one or multiple data variations between the source and target domains, e.g., channel realization, blockages, the frequency band used, etc.
- a distribution shift between the source and target conditional distributions, i.e., $P^s_{\mathsf{X}|\mathsf{Y}} \neq P^t_{\mathsf{X}|\mathsf{Y}}$, as depicted in Fig. 3(d). This shift is often considered unchanged (i.e., $P^s_{\mathsf{X}|\mathsf{Y}} = P^t_{\mathsf{X}|\mathsf{Y}}$) to ensure that the label random variable Y causes the input random variable X in the same way between the source and target domain. Because $P_{\mathsf{X}|\mathsf{Y}} = P_{\mathsf{X}|\mathsf{Y}}\,P_{\mathsf{X}}/P_{\mathsf{Y}}$, this shift in (2) occurs when at least one of the covariate shift or the concept shifts are present, assuming that P_{Y} is unchanged.

Note that each type of distribution shift is often studied independently, and the existing algorithms for DG assume that the other shifts are not present [21]. It is worth mentioning that most proposed algorithms in the literature focus on the covariate shift only and are specialized in classification tasks.

DG is closely related to other generalization concepts such as: multi-task learning [22], transfer learning [5], zero-shot learning [23], domain adaptation [6], and test-time training [24]. Figure 4 illustrates the taxonomy of these generalization concepts and we will subsequently elaborate further on their differences.

C. Related Concepts to DG

When the source and target domains are assumed to be the same (i.e., $P^s_{\mathsf{X},\mathsf{Y}} = P^t_{\mathsf{X},\mathsf{Y}}$), the learned model is not exposed to any domain shift. This assumption is pervasive in wireless communication ML applications where training and test datasets are usually generated using the *same* system model and/or assumed to originate from the same propagation environments.[3] However, in practice, this assumption is often violated and the test domains are usually different from the training domain(s). Supervised and multi-task learning are two common learning techniques where no domain shift occurs.

Supervised learning learns a mapping between inputs and outputs assuming that training and test samples are identically and independently distributed (i.i.d). Let M be the number of distinct source domains. Supervised learning considers a *single* domain (M=1). Because supervised learning does not handle domain shifts, the source and target samples are i.i.d. drawn from the same joint distribution, i.e., $P^s_{\mathsf{X}\mathsf{Y}} = P^t_{\mathsf{X}\mathsf{Y}}$. This is different from the DG setting where the i.i.d assumption is violated since the source and target samples are drawn from different distributions.

Multi-task learning trains a single model to simultaneously perform multiple tasks, i.e., $M > 1$. For the rest of the paper, a task refers to a type of problem to be solved such as classification or regression. Different tasks result in different but related domains or datasets which enable learning shared representations between tasks. Note that each task is characterized by a specific joint distribution $P^{s_i}_{XY} \neq P^{s_j}_{XY}$ for $i \neq j$ with

[3]By propagation environments, we refer to all the different parameters that impact the signal propagation conditions like path loss, coherence time, blockages, etc.

Authorized licensed use limited to: Northeastern University. Downloaded on February 26,2024 at 14:58:14 UTC from IEEE Xplore. Restrictions apply.

$1 \leq i, j \leq M$. The objective of multi-task learning is to learn a model that performs well on the source tasks, meaning that target and source domains are the same. This is to be opposed to DG which aims to generalize to unseen domains/tasks.

When the source and target domains are not identical, i.e., $P^s_{\mathsf{X},\mathsf{Y}} \neq P^t_{\mathsf{X},\mathsf{Y}}$, the domain shift problem has to be addressed. In practice, DNNs trained on a given source domain suffer significant performance degradations on a different target domain even when the latter covers small variations compared to the source domain. In such cases, it is important to determine the origin and the assumptions behind the domain shift. As shown in Fig. 4, we overview four generalization paradigms in addition to DG. One important differentiating aspect between these paradigms is the *access to the target data during training*. Domain adaptation and transfer learning both assume that test data is available which is not the case in DG and zero-shot learning.

Transfer learning seeks to transfer the knowledge learned from one or multiple source domains/tasks to a different but related one. Finetuning is a common transfer learning technique where a model is first pre-trained on (often large) source datasets and then finetuned on different target datasets. The key difference between transfer learning and DG is access to the target data. Since transfer learning techniques involve model finetuning, target samples are required. However, DG assumes that target data is not accessible during training. In other words, transfer learning requires target data to generalize via additional training whereas DG seeks to achieve generalization on the target domain without any additional training. Nonetheless, both transfer learning and DG consider the target and source distributions to be different. Specifically, transfer learning usually deals with different label spaces which is also the case in heterogeneous DG.

Domain adaptation considers the target marginal to be accessible at training time and hence can leverage target samples to improve the performance of DNN on target domains at inference time. The default setting of domain adaptation assumes that target samples are unlabelled and the target and source domain share the same label space (i.e., homogeneous domain adaptation). Different from domain adaptation, DG restricts the training to samples from the source domains only. As in the heterogeneous DG setting, heterogeneous domain adaptation extends the original setting by allowing for different label spaces between the source and target domains. Other domain adaptation variants are proposed in the literature such as zero-shot domain adaptation where target data are not required during training [25].

Zero-shot learning primarily deals with label space shift (i.e., $\mathcal{Y}^s \neq \mathcal{Y}^t$). In other words, the goal of zero-shot learning is to recognize or generalize to classes or target values that are unseen during training. However, DG was initially proposed to handle the covariate shift arising from a change in the marginal distribution only (i.e., $P^s_X \neq P^t_X$). Note that zero-shot learning is related to heterogeneous DG since both covariate and label space shifts are allowed.

Test-time adaptation/training aims to adapt a trained model to new domains without accessing the source data and human annotations at test time. Test-time training methods train a model to perform two tasks, namely the main task and the self-supervised auxiliary task. The self-supervised task will be used at test time to create labels for the unlabeled test samples. The standard version of test-time training requires a very limited amount (e.g., a mini-batch) of data to finetune the model based on the auxiliary task. This is where test-time training differs from DG due to the use of test data for updating the model parameters.

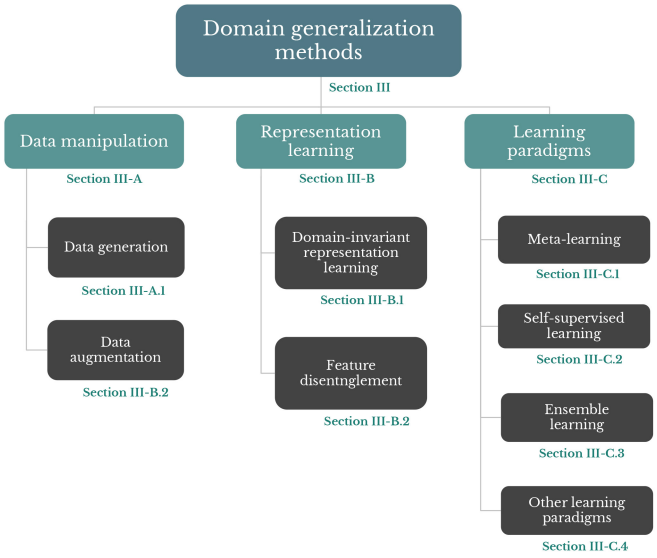


Fig. 5. Taxonomy of domain generalization methods.

Continual/lifelong learning learns a model on multiple domains or tasks sequentially without forgetting the knowledge previously learned. Continual learning assumes that the model does not have access to data from previous tasks and updates the parameters using labeled data from new tasks or domains. This is different from DG where the objective is to generalize to new domains without accessing target data or finetuning the model on the target domain.

III. DG Methods

In this section, we put forward the current state-of-the-art methodologies for handling DG. We also specify which methods cope with distribution shifts beyond the covariate shift. Fig. 5 presents the organization of the covered DG methodologies across the next three sections.

A. Data Manipulation

In order to generalize to unseen scenarios, this category of methods manipulates the DNN input data. Two types of manipulations are possible either in the raw input space or in the latent input space: *i)* data augmentation by adding random noise or transformation to the input data, and *ii)* data generation which generates new training samples using generative models. The main objective of these methods is to increase the quantity and improve the diversity of the training dataset for better generalization capabilities without requiring manual labeling of datasets.

A data manipulation operation is represented by an arbitrary function $\mathcal{M}(\cdot)$ which transforms the input data X to the

Authorized licensed use limited to: Northeastern University. Downloaded on February 26,2024 at 14:58:14 UTC from IEEE Xplore. Restrictions apply.

manipulated data $X' = \mathcal{M}(X)$. Given a DNN that is represented as an input-output function $g(\cdot)$, the learning objective of data manipulation for DG can be expressed as follows:

$$\min_{g} \underbrace{\mathbb{E}_{\mathbf{X},\mathbf{Y}}\big[\mathcal{L}(g(X), Y)\big]}_{\text{task loss}} + \underbrace{\mathbb{E}_{\mathbf{X}',\mathbf{Y}}\big[\mathcal{L}\big(g\big(X'\big), Y\big)\big]}_{\text{data manipulation loss}}, \quad (3)$$

where $\mathcal{L}(\cdot,\cdot)$ is the DNN cost function. It is worth noting that most data manipulation techniques proposed in the literature are geared towards computer vision applications where all datasets consist of images. In this section, we describe these methods within the context of vision applications and point out their potential use for wireless applications.

1) Data Generation: Generating new data samples using generative models is a popular technique to augment existing datasets so as to cover richer training scenarios, thereby enhancing the generalization capability of a DNN. The data manipulation function $\mathcal{M}(\cdot)$ in (3) can be represented by deep generative models such as variational auto-encoder (VAE) [26] and generative adversarial network (GAN) [27].

Various distribution distance metrics can be employed to generate high-quality samples including:

- *domain discrepancy measures* such as the maximum mean discrepancy (MMD) [28] to minimize the distribution divergence between real and generated data samples.
- *the Wasserstein distance* between the prior distribution of the DNN input and a latent target distribution as carried out in Wasserstein auto-encoder (WAE) [29]. This metric is a regularization that encourages the encoded training distribution of a WAE to match the data prior and hence preserves the semantic and domain transfer capabilities.
- *semantic consistency loss functions* that maximize the difference between the source and the newly generated distributions, thereby creating new domains that augment the existing source domains [30].

It is also possible to generate new domains instead of new data samples using adversarial training [31] where one or multiple generative models are trained to progressively generate unseen domains by learning relevant cross-domain invariant representations. Such an alternative involves an entire generative model pipeline composed of multiple DNNs trained in cascade or in parallel, and therefore has a significant computational cost. As one example for channel estimation problems, one can start by generating line-of-sight datasets and then progressively increase the rank of the estimated MIMO channel to multi-path models up to full-rank channels such as rich-scattering MIMO channels.

Furthermore, the data manipulation function $\mathcal{M}(\cdot)$ can also be defined without training generative models. In particular, it is possible to generate new data samples by linearly interpolating any two training samples and their associated labels as done in the low-complexity Mixup method [32]. More recently, many techniques have built upon Mixup for DG to *i)* generate new data samples by interpolating either in the raw data space [33], [34], [35], or *ii)* to build robust models with better generalization capabilities by interpolating in the feature space [36], [37], [38].

2) Data Augmentation: DNNs are heavily reliant on large datasets to enhance the generalization by avoiding overfitting [39]. Data augmentation methods provide a cheap way to augment training datasets. They artificially inflate the dataset size by transforming existing data samples while preserving labels. Data augmentation includes geometric and color transformations for visual tasks, random erasing and/or permutation, adversarial training, and neural style transfer. These data augmentation techniques are highly geared towards computer vision and natural language processing applications. However, they can be tailored to wireless communications applications due to the inherent properties of digital constellations. For instance, data augmentation can account for specific rotations applied to constellation symbols yielding other valid symbols, leading to constellation-preserving augmentation. Another example exploits the distortion within one cluster of estimated symbols and then replicates it in another cluster of estimated symbols. By doing so, new realizations of the distortion are synthesized and used with another symbol center to create a new observation. The benefit of communications-tailored augmentations has been reported in [40]. In this section, we highlight data augmentation methods that are still unexplored by the communication community despite being reported beneficial for domain generalization by the machine learning community.

In general, every data augmentation operation can be considered as a data manipulation function $\mathcal{M}(\cdot)$ in (3). Here, we classify the data augmentation methods for DG into two categories:

- *domain randomization:* this family of methods creates a variety of datasets stemming from data generation processes (e.g., simulated environments) with randomized properties and trains a model that generalizes well across all of them.
- *adversarial data augmentation:* this family of methods guides the augmentation by enhancing the diversity of the dataset while ensuring their reliability for better generalization capabilities.

a) Domain randomization: The reality gap between the data domains resulting from simulations and real-world data collections often leads to failure due to distribution shifts. This gap is triggered by an inconsistency between the physical parameters of simulations (e.g., channel distribution, noise level) and, more fatally, the incorrect physical modeling (e.g., physical considerations of wireless communication [41], [42]). To perceive how DNNs should be trained and evaluated under data distribution shifts for communication applications, Fig. 6 depicts the training and evaluation pipeline where datasets are generated through communication systems models. There, it is seen that source (i.e., training) and target (i.e., test) domains, $\mathcal{D}^{\text{train}}$ and $\mathcal{D}^{\text{test}}$, are obtained according to the training and test scenarios, $\mathcal{S}^{\text{train}}$ and $\mathcal{S}^{\text{test}}$. The latter are determined by defining a set of communication scenarios by varying one or multiple communication parameters of interest. The choice of these parameters dictates the data domains and hence provides a way to control and then analyze the impact of distribution shifts on the performance of DNNs. For instance, research efforts to design broadband ML-aided

Authorized licensed use limited to: Northeastern University. Downloaded on February 26,2024 at 14:58:14 UTC from IEEE Xplore. Restrictions apply.

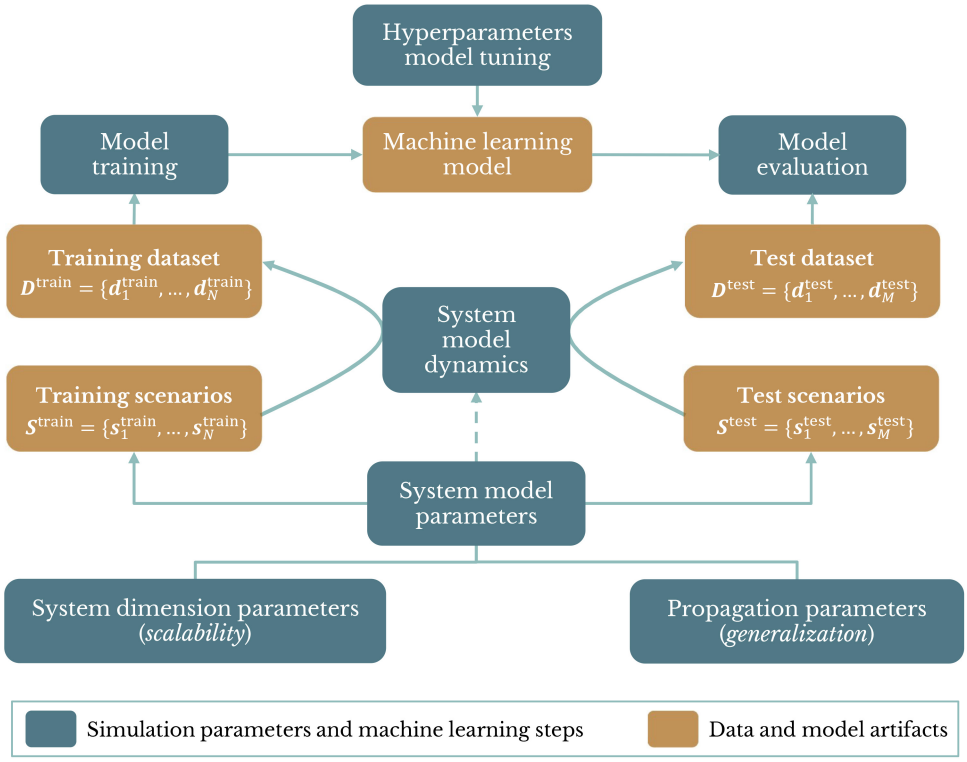


Fig. 6. Summary of the training and evaluation pipeline of machine learning models under data distribution shifts for communication applications.

decoding algorithms should vary the signal frequency and assess the generalization capability of DNNs when trained on carriers in the sub-6 GHz band and then evaluated on a different communication band.

Domain randomization generates new data samples stemming from simulated dynamics of complex environments. For computer vision applications, the function $\mathcal{M}(\cdot)$ in (3) encloses different manual transformations such as altering object properties (e.g., shape, location, texture), scene editing (e.g., illumination, camera view), or random noise injection [43]. For real-valued data input vectors, augmentation involves scaling, pattern switching, and random perturbation [44]. These augmentation methods are particularly interesting for wireless communication applications because they handle general signal transmission scenarios that are tolerant to variations in the path-loss coefficient, synchronization delays, signal-to-noise ratio, etc.

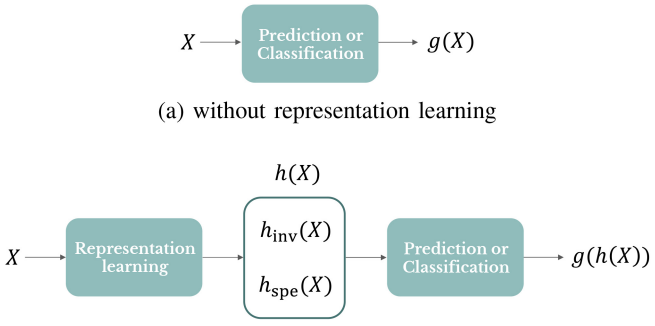
b) Adversarial data augmentation: The fact that most domain randomization described in Section III-A2a is performed randomly indicates that there exist potential improvements to remove ineffective randomization that does not help with DNNs' generalization. This optimization is performed by adversarial data augmentation.

Toward this goal, research efforts have been dedicated to designing better strategies for non-random data augmentation. By modeling the dependence between the data sample X, its label Y, and the domain label d (cf. Definition 2), it has been shown that the input data can be perturbed along the direction of greatest domain change (i.e., domain gradient) while changing the class label as little as possible [45]. Another line of work devised an adaptive data augmentation procedure where adversarially perturbed samples in the feature space are iteratively added to the training dataset [46]. It is also possible to train a dedicated transformation network for data augmentation by *i)* maximizing the domain classification loss on the transformed data samples to tolerate domain generation differences, and *ii)* minimizing the label classification loss to ensure that the learned augmentation does not affect the DNN performance [47]. While adversarial data augmentation can provide richer datasets and fill in data gaps against some adversarial examples, this comes at the cost of a more complex training procedure which is known to be less stable and computationally extensive.

When it comes to wireless communications applications, physics-based models are available to guide data augmentations that are consistent with the law of physics, beyond purely random strategies. For example, the study of the achievable rate of reconfigurable intelligent surface (RIS)-aided communication systems exhibit the same performance regardless of the carrier frequency due to the scaling invariance property of Maxwell's equations when no source is present (i.e., passive RISs) [48]. Another interesting implication stemming from the symmetry of Maxwell's equations is the frequency independence property of certain wideband antennas that display very similar radiation pattern, gain, and impedance above a certain threshold frequency [49]. This suggests that the generation of wireless datasets for far-field communication can be made independent of the carrier frequency for specific types of antennas.

From this perspective, data augmentation methods that are aware of the physics of wave propagation do not blindly generate source and target domains for different carrier

Authorized licensed use limited to: Northeastern University. Downloaded on February 26,2024 at 14:58:14 UTC from IEEE Xplore. Restrictions apply.



(a) without representation learning

(b) with representation learning

Fig. 7. Illustration of ML-aided classification/prediction (a) without an explicit representation learning step (a.k.a. end-to-end learning), and (b) with a representation learning step.

frequencies. They should instead collapse the data augmentation process to scenarios that do enjoy the scaling invariance property. As a result, not only do data augmentation techniques become efficient but also physically consistent with the electromagnetic properties of RISs.

B. Representation Learning

Generalizing to unseen scenarios is not solely dependent on the DNN prediction approximation function $g(\cdot)$ given in (3). It also depends on the data representations (i.e., features) learned by the DNN [50]. To better isolate these two distinct tasks, one can view the overall DNN approximation function, $g(h(\cdot))$, as a composition of a prediction/classification function $g(\cdot)$ and a representation learning function $h(\cdot)$. Fig. 7(b) depicts this decomposition, $h(X)$, as the output of the representation learning step. In theory, this representation in the feature space comprises two separate representations. The first one denoted by $h_{\text{inv}}(X)$ is a domain-invariant representation that is shared across domains (a.k.a., cross-domain representation) and is key to enabling generalization over multiple domains. The second representation $h_{\text{spe}}(X)$, however, is domain-specific and represents the variation pertaining to a specific domain. In practice, these two representations can either be non-separable or separable. For instance, several earlier research works [51], [52], [53] have shown that in the Fourier spectrum of signals, the phase component predominantly carries low-level statistics whereas the amplitude component mainly contains high-level semantics. Hence, Fourier phase features represent domain-invariant features that cannot be easily affected by domain shifts when used for DG [54].

From a mathematical point of view, the optimization problem of representation learning can be written as follows:

$$\min_{g,h} \underbrace{\mathbb{E}_{\mathsf{X},\mathsf{Y}}\big[\mathcal{L}(g(h(X)),Y)\big]}_{\text{task loss}} + \lambda \underbrace{r(X)}_{\text{regularization loss}}, \quad (4)$$

where $r(X)$ is a regularization function and λ is the associated regularization parameter.

Depending on the type of the regularization function $r(X)$ or the representation learning function $h(\cdot)$, it is possible to categorize representation learning for DG into two categories:

- *domain-invariant representation learning:* the goal of this family of methods is to learn features that are invariant across different domains. These features are transferable from one domain to another, hence their importance for domain generalization.
- *feature disentanglement:* these methods decompose a feature representation into one or multiple sub-features, each of which is either domain-specific or domain-invariant.

1) Data-Invariant Representation Learning:

a) Kernel-based methods: Learning representation using kernel methods (e.g., support vector machines [55], kernel component analysis [56]) is a classical problem in the ML literature. In such a setting, the representation learning function $h(\cdot)$ in (4) maps the data samples to the feature space using kernel functions (e.g., radial basis function (RBF), Gaussian, and Laplacian kernels).

For domain generalization, several methods were devised to learn domain-invariant kernels to determine $h(\cdot)$ from the training dataset. Specifically, a positive semi-definite kernel learning approach for DG was proposed in [57] by considering the conventional supervised learning problem where the original feature space is augmented to include the marginal distribution that generates the features. It is also possible to learn kernel functions by minimizing the distribution discrepancy between all the data samples in the feature space. This method is known as domain-invariant component analysis (DICA) [58] and is one of the classical kernel methods for DG.

For classification tasks, in the presence of covariate shift only, a randomized kernel algorithm was devised in [59] to extract features that minimize the difference between the marginal distributions across domains. Multi-domain discriminant analysis (MDA) and scatter component analysis (SCA) approaches were proposed in [60], [61] to learn a domain-invariant feature transformation in presence of both covariate and conditional shifts across domains. This is done by jointly minimizing the divergence among domains within each class and maximizing the separability among classes.

b) Domain adversarial learning: Since the presence of spurious features in the data decreases the robustness of DNNs, adversarial learning is a widely used technique to learn invariant features by training generative adversarial networks (GANs). Specifically, the discriminator is trained to distinguish the domains while the generator is trained to fool the discriminator so as to learn domain invariant feature representations for DG [62]. Another line of work in [63] generated a continuous sequence of intermediate domains flowing from one domain to another to gradually reduce the domain discrepancy, and hence improve the DNN generalization ability on unseen target domains. Learning class-wise adversarial networks for DG was also proposed in [64] based on conditional invariant adversarial training when both covariate and conditional shifts coexist.

c) Explicit feature alignment: This family of methods learns domain-invariant representations by aligning the features across source domains using one of the following two mechanisms:

- explicit feature distribution alignment through distance minimization or moment matching.

Authorized licensed use limited to: Northeastern University. Downloaded on February 26,2024 at 14:58:14 UTC from IEEE Xplore. Restrictions apply.

- feature normalization addressing data variations to avoid learning nonessential domain-specific features.

Feature distribution alignment methods were devised to impose a variety of distribution distances such as the maximum mean discrepancy (MMD) on latent feature distributions [62], [65], and the label similarities for samples of the same classes from different domains using the Wasserstein distance [66]. Moment matching for multi-source domain adaptation (M3SDA) was also introduced in [67] to transfer learned features from multiple labeled source domains to an unlabeled target domain by dynamically aligning moments of their feature distributions.

Feature normalization methods, however, focus on increasing the discrimination capability of DNNs. They do so by normalizing the features to eliminate domain-specific variation while keeping domain-invariant features to enhance generalization. In particular, instance normalization (IN) [68] and batch instance normalization (BIN) [69] have been proposed to enhance the generalization capabilities of convolutional neural networks (CNNs). Instance normalization has been applied in [70] for DG where labels were missing in the training domains to acquire invariant and transferable features. It was also shown that adaptively learning the normalization technique can improve DG without predefining the normalization technique in the DNN architecture a priori [71].

d) Invariant risk minimization: Another unique perspective on learning domain-invariant representations for DG is to constrain DNNs to have the same output across all domains. The motivation behind this constraint is that an optimal representation for prediction or classification is *the cause* of the DNN output label. This causal relationship from the representation (i.e., the cause) to the label (i.e., the effect) should not be affected by other factors including the domain input. Therefore, the optimal representation is domain invariant and can be learned using invariant risk minimization (IRM) [72]. Given K different domains, the IRM problem can be formulated as follows:

$$\min_{h\in\mathcal{H}} \sum_{k=1}^{K} \mathbb{E}_{\mathsf{X}_k,\mathsf{Y}_k}\left[\mathcal{L}(g(h(X_k)),\, Y_k)\right] \tag{5a}$$

$$\text{subject to } g \in \bigcap_{k=1}^{K} \arg\min_{g'\in\mathcal{G}} \mathbb{E}_{\mathsf{X}_k,\mathsf{Y}_k}\left[\mathcal{L}\Big(g'(h(X_k)),\, Y_k\Big)\right], \tag{5b}$$

where $\mathcal{H}$ and $\mathcal{G}$ are the learnable function classes for representation and task functions, $h(\cdot)$ and $g(\cdot)$, respectively. The optimization in (5) finds the optimal representation function $h(\cdot)$ that minimizes the sum of all the task losses in (5a) given in (4). This minimization is carried out under the constraint in (5b) which ensures that all domains share the same optimal representation function $h(\cdot)$.

The idea behind the IRM formulation has drawn significant attention. Specifically, the IRM optimization was extended to text classification [73], reinforcement learning [74], self-supervised settings [75], and to the case of extrapolated task losses among source domains [76]. Moreover, it was shown in [77] that constraining the invariance to the task function $g(\cdot)$ only — as done in (5) — is not enough to guarantee the causal relationship from the representation to the label. A new regularization has thus been proposed to ensure that the representation function $h(\cdot)$ cannot capture fully invariant features that break down the assumed causality as required by the IRM formulation.

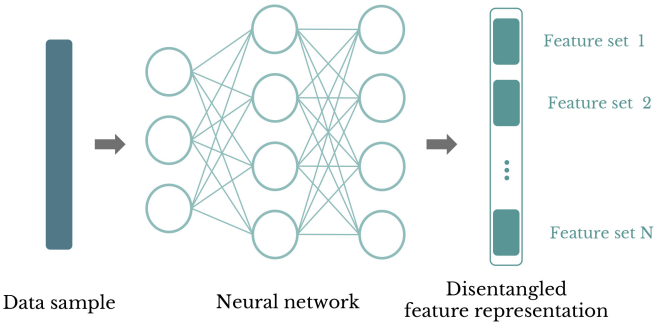


Fig. 8. Illustration of how a trained neural network transforms a data sample into a disentangled representation vector that factorizes into N small feature vectors.

2) Feature Disentanglement: Unlike domain-invariant representation learning, disentangled representation learning relies on DNNs to learn a function that maps a data sample to a feature vector, which factorizes into distinct feature sets as depicted in Fig. 8. There, it is seen that the entire feature space can be decomposed into a set of feature subspaces. Each feature set is a representation pertaining to a specific feature subspace only. When the feature representation is decomposable into multiple non-overlapping feature subsets, the feature representation is said to be "disentangled".

The importance of disentanglement-based representation learning for DG stems from the fact that features can be explicitly decomposed into domain-invariant and domain-specific features. As a result, the representation function $h(\cdot)$ defined in (4) can be decomposed into two distinct representation functions: $h_{\text{inv}}(\cdot)$ for domain-invariant representation and $h_{\text{spe}}(\cdot)$ for domain-specific representation. The disentanglement-based optimization can be formulated as follows:

$$\min_{h_{\text{spe}},\, h_{\text{inv}},\, g} \quad \underbrace{\mathbb{E}_{\mathsf{X},\mathsf{Y}}\left[\mathcal{L}(g(h_{\text{inv}}(X)),\, Y)\right]}_{\text{task loss}} + \lambda \underbrace{r(X)}_{\text{regularization loss}} + \mu \underbrace{\mathbb{E}_{\mathsf{X}}\left[\mathcal{L}\big(h_{\text{inv}}(X), h_{\text{spe}}(X), X\big)\right]}_{\text{reconstruction loss}}, \tag{6}$$

where λ and μ are regularization parameters. In (4), the regularization loss encourages the separation between domain-invariant and domain-specific features, while the reconstruction loss ensures that such separation does not lead to significant information loss. In other words, regularization and reconstruction losses are competing penalties that add up to the task loss, and it is the task of the ML designer to find the suitable trade-off that enhances the generalization of DNNs.

a) Multi-component analysis: Multi-component methods dedicate different sets of parameters to learn domain-invariant and domain-specific features. The method "UndoBias" proposed in [78] learns dedicated SVM models. It represents the dedicated SVM parameters, $\boldsymbol{w}_k$, pertaining to the kth domain as a perturbation of the domain-invariant parameters

Authorized licensed use limited to: Northeastern University. Downloaded on February 26,2024 at 14:58:14 UTC from IEEE Xplore. Restrictions apply.

$\boldsymbol{w}$ with the domain-specific parameters $\Delta \boldsymbol{w}_k$, i.e., $\boldsymbol{w}_k = \boldsymbol{w} + \Delta \boldsymbol{w}_k$. This method has been extended for multi-view vision tasks by introducing a regularization to minimize the mismatch between any two view representations [79] for better generalization. Neural networks have also been used to capture disentangled representations by learning domain-specific networks for each domain and one domain-invariant network for all domains [80]. Another line of work considered manually comparing specific areas of DNN's attention heatmaps from different domains which proved beneficial to learning disentangled representations and ensuring a more robust generalization [81].

b) Generative modeling: Generating data samples whose feature representations are disentangled requires adapting the data generative process of generative models to new constraints. The latter can be incorporated in the loss functions of GANs to encourage feature disentanglement by separating the domain-specific and domain-invariant features [82]. An autoencoder-based variational approach was devised to disentangle the features by learning three independent latent subspaces, one for the domain, one for the class, and one for any residual variations [83]. To generate domains that are different from the source domain, the discrepancy between augmented and sources domains was maximized for out-of-domain augmentation using meta-learning under a semantic consistency constraint [84].

For classification tasks, diversifying the inter-class variation by modeling potential seen or unseen variations across classes was formulated as a disentanglement-constrained optimization problem [85]. This was made possible by minimizing the discrepancy of the inter-class variation where both intra- and inter-domain variations are regarded as constraints.

C. Learning Paradigms

Aside from data manipulation and representation learning, DG has also been investigated within broader machine learning paradigms such as meta-learning, self-supervised learning, ensemble learning and other learning strategies.

1) Meta-Learning: Meta-learning [86], also known as learning-to-learn is a research area that has attracted much interest in recent years. The main goal of meta-learning is to learn a general model using samples from multiple tasks to quickly adapt to new unseen tasks. The learned meta-model encompasses the general knowledge from all the different training tasks which makes it a better model initialization to adapt for new tasks [87].

Traditional supervised learning (SL) methods learn a model f_θ that maps inputs to outputs. The model's parameters θ are learned by minimizing a loss function given a dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{m}$ as follows:

$$\theta^*_{\text{SL}} = arg\ min_\theta \mathcal{L}(\mathcal{D}, \theta)$$

At each iteration, the parameters are updated based on a specific optimization procedure g_ω where ω denotes all the pre-defined assumptions about the learning algorithm such as the function class of f (e.g., DNN), the initial model initialization, the choice of the optimizer, etc. In the literature, ω is also called *pre-defined meta-knowledge* [88]. It is straightforward to observe that the model's performance depends drastically on ω.

In addition, it is common to split the dataset $\mathcal{D}$ into training and testing sets. The model is first learned using the training samples, and the generalization of the model is subsequently evaluated on the test set with unseen samples and known outputs. Consequently, the learned parameters θ_{SL} are specific to the dataset $\mathcal{D}$ and are not guaranteed to generalize to samples different from the ones in $\mathcal{D}$.

Different from the supervised learning setting, meta-learning aims to learn a meta-knowledge ω over a distribution of tasks $p(\mathcal{T})$. A task i can be defined by a loss function and a dataset (i.e., $\mathcal{T}_i = \{\mathcal{L}_i, \mathcal{D}_i\}$). Learning the meta-knowledge from multiple tasks enables the quick learning of new tasks from $p(\mathcal{T})$. In meta-learning, different choices of the meta-knowledge ω are proposed such as parameter initialization, optimizer, hyperparameters, task-loss functions, etc. We refer the interested reader to [88] for a detailed discussion about the different choices for ω.

Meta-learning algorithms also involve two stages, namely meta-training followed by meta-testing. The objective of meta-training is to learn the "best" meta-knowledge ω across multiple tasks. To do so, a set of training tasks $\mathcal{T}_{\text{train}} \sim p(\mathcal{T})$ is used where each task i has training and validation datasets (i.e., $\mathcal{D}_i = \{\mathcal{D}_i^{\text{train}}, \mathcal{D}_i^{\text{val}}\}$). The meta-training phase is commonly presented as a bi-level optimization problem [88] as follows:

$$\omega^* = \overbrace{\arg\min_{\omega} \sum_{i=1}^{|\mathcal{T}_{\text{train}}|} \mathcal{L}\left(\theta_i^*(\omega), \mathcal{D}_i^{\text{val}}\right)}^{\text{outer level}}, \tag{7}$$

$$\text{s.t.} \quad \underbrace{\theta_i^*(\omega) = \arg\min_{\theta} \mathcal{L}_i\left(\mathcal{D}_i^{\text{train}}, \theta, \omega\right)}_{\text{inner level}}. \tag{8}$$

The inner level consists in learning task-specific learners conditioned on the meta-knowledge ω. Note that the inner level only optimizes the task-specific parameters θ using the task train datasets $\mathcal{D}_i^{\text{train}}$ and does not change ω. Whilst, the outer level learns ω that minimizes the aggregated losses from all the train tasks on their validation datasets.

In the literature, it is common to divide meta-learning methods into three families: optimization-based, model-based, and metric-based. Optimization-based methods, promoted by the Model Agnostic Meta-Learning (MAML) algorithm [87] have been recently adopted for domain generalization. The general idea is to consider the different domains as different tasks. Hence, data from multiple source domains are divided into meta-training and meta-testing sets. By training with data from different domains, the meta-learner is exposed to domain shift and is required to learn a meta-knowledge that quickly adapts to domain shift in new unseen domains [89].

2) Self-Supervised Learning: Self-supervised learning (SSL) is a learning paradigm that generates labels from data and subsequently uses these labels as ground truth. SSL is useful in real-world applications where abundant unlabeled data is available, especially when the labeling

Authorized licensed use limited to: Northeastern University. Downloaded on February 26,2024 at 14:58:14 UTC from IEEE Xplore. Restrictions apply.

process is cumbersome and expensive. Another motivation behind SSL is to learn rich and general representations, unlike supervised learning methods that learn biased representations via the supervision signal or the type of annotations [90]. In supervised learning, labels serve as the supervision signal to learn a specific task. However, in SSL, a model is learned using the data as a supervision signal. In other words, the labels in SSL are generated from the data itself. The SSL pipeline can be divided into two parts:

- learn feature representations by solving a *pretext* task. An example of a pretext task is to retain part of the input data to be predicted by a model that is trained on the other part of the data [91]. Another pretext task consists of learning the relationship between data instances (e.g., similarity) or reconstructing an input from its shuffled parts (also known as the jigsaw puzzle). Note that the labels (or supervision signal) for the pretext task is generated from the input data, thus no human intervention is needed;
- solve a downstream task using the learned representations and a few annotated data.

SSL is applied in DG to learn domain-invariant features that help avoid overfitting on domain-specific biases while aligning features from different source domains. As discussed in Section III-B1, these invariant features can be leveraged in unseen target domains to achieve better generalization [58]. In this context, contrastive learning is a well-known SSL method that aims to learn latent representations such that positive instances are close and negative samples are pushed away. Therefore, in the learned embedding space, the distance between similar instances is reduced while the distance between negative pairs is increased. For instance, the authors in [92] proposed two self-supervised contrastive losses to measure feature dissimilarities in the embedding space. For dissimilarities across domains, the authors used a Mix-up layer [32], [34] (i.e., a convex combination of samples' embeddings from different domains) to compute the interpolated feature representation across domains. Thus, the regularisation loss is defined as the distance between the individual representations and the interpolated one using the mix-up layer. One caveat of this method is that it assumes the label space does not change for all the domains.

3) Ensemble Learning: Ensemble learning [93] is a famous technique in traditional and modern machine learning where multiple models are learned and combined for prediction/classification. The same idea was also exploited for DG. The most straightforward approach is to learn a model for each source domain and average the individual predictions to compute the final ensemble prediction [94], [95]. Instead of learning separate models for each source domain, it is common to design the ensemble as a shared feature extractor and different domain-specific heads [94]. Another line of work focuses on weighting the individual models' predictions. For instance, the domain-specific models can be weighted differently depending on the similarity of the target domain to the source domain. The authors in [96] proposed to learn a domain predictor that predicts the probability that a target sample belongs to a source domain. These probabilities can be used to fuse the models' predictions at test time.

An alternative solution proposes to train domain-invariant classifiers for each source domain by learning domain-specific normalization [97], [98]. All the classifiers share the same parameters except the ones in the normalization layers. The objective of learning domain-specific normalization is to obtain domain-agnostic latent feature space that can be used to map samples from unknown domains to the source domains. This idea is related to the feature alignment methods reviewed in Section III-B1c.

Alternatively, the stochastic weight averaging (SWA) method [99] aggregates weights at different training epochs to form an ensemble model instead of combining the predictions of multiple learners. Starting from a pre-trained model, SWA trains a single model using a cyclic learning rate schedule (or a constant high learning rate) and saves model snapshots corresponding to different local minima. Averaging these points leads to better solutions in flatter regions of the loss landscape. Intuitively, flatter minima are more robust than sharp minima to changes in the loss landscape between the training and testing datasets [99]. Consequently, this weight averaging idea was extended to the DG proving that flat minima lead to better generalization on unseen domains [100].

4) Other Learning Paradigms: Other learning-based approaches are proposed for DG. For instance, hypertnetwork-based learning [101] is an approach that learns a network (i.e., the hypernetwork) to generate weights for another network called the main network. The latter represents the usual model that maps raw data to their targets or labels. The goal of the hypernetwork is to generate a specific set of weights depending on inputs about the structure of the weights or tasks. Different from the usual supervised learning setting, only the hypernetwork's parameters are learned during training whilst keeping the main network's parameters unchanged. At inference, the main network is evaluated based on the weights generated by the hypernetwork. Recent work proposed hypernetwork-based algorithms for DG in natural language processing [102] and vision [103]. For vanilla DG, a straightforward application of hypernetworks is to train a hypernetwork on data samples from different source domains to produce the model's weights for each domain. On the other hand, for compound DG, the appropriate approach is to first learn a latent embedding space for the different domains, then the hypernetwork learns to map the latent features to a set of weights so as to compute model predictions.

Another line of work is gradient-based methods which capture invariance in the gradient space [104], [105]. Intuitively, gradient matching methods strive to align the gradients across different domains by enforcing the network parameters to contribute similarly for different domains. For instance, Invariant Learning Consistency (ILC) measure [104] matches Hessians across different domains after convergence assuming that the Hessian matrices are diagonal. Similarly, the Inter Gradient Alignment (IGA) algorithm is proposed to minimize the variance of average gradients over domains [105]. Further recent works investigated how the Hessian alignment can improve the transferability between domains [106].

Authorized licensed use limited to: Northeastern University. Downloaded on February 26,2024 at 14:58:14 UTC from IEEE Xplore. Restrictions apply.

In the next sections, we will overview the different applications of the techniques detailed above to wireless communication problems.

IV. Domain Generalization Applications in Wireless Communications

When designing data-driven ML-based algorithms for solving wireless communication problems, it is crucial to ensure that the developed algorithms have guaranteed generalization capabilities. However, little effort has been devoted to investigating the DG issue despite the huge research effort in applying data-driven machine learning techniques to various wireless communication applications. The goal of this section is to overview the existing DG methodologies that were applied by the communication community and summarize the learned lessons from their applications.

A. Channel Coding

Channel codes are guaranteed to be optimal only when the block-length approaches infinity [107]. Under short and moderate block length regimes, however, the potential of DNNs to bring further encoding improvements has been investigated using autoencoders [108], recurrent DNNs [109], and diffusion models [110]. At the time of writing, the only significant work in domain generalization for channel coding is the meta-learning benchmark developed in [111]. There, five channel families were considered, namely, AWGN, bursty, memory noise, and multipath interference channels as well as a real-world channel measurement based on software-defined radio (SDR). Each channel model corresponds to a meta-task. The distribution shift between the train and test distribution tasks was quantified using the Kullback–Leibler (KL) divergence to assess the impact of training distribution diversity on the meta-learning performance. The encoding performance has been evaluated by assessing the channel decoding accuracy. This evaluation protocol is relevant in practice because communication standards defining the encoding protocol are not easy to change. However, decoders can be upgraded without changing the standard. The authors reported high generalization capabilities across some channel models (e.g., from multipath to AWGN model) but not for other channel model pairs (e.g., bursty to AWGN). Overall, the authors confirmed that robustness to distribution shift across encoding channel models is an issue for both non-adaptive and meta-learned adaptive decoders.

B. Channel Decoding

Iterative turbo/LDPC decoders [112], [113] based on the belief-propagation (BP) framework [114] are recognized as state-of-the-art channel decoders because of their capacity approaching/achieving performance for relatively large block lengths. For this reason, they have been adopted in the 4G/5G communication standards.

Many deep learning studies have shown that data-driven ML techniques can decrease the BP decoding complexity especially for short-to-moderate block lengths [115]. For short-block-length polar codes, [116] (e.g., 16 bits), DNN-based decoders were shown to exhibit near-optimal performance using maximum a posteriori (MAP) decoding [117]. For larger block length codes (i.e., larger than 100 bits), the BP algorithm was unfolded into a DNN in which weights are assigned to each variable edge, thereby showing an improvement in comparison to the baseline BP method [118]. By varying the signal-to-noise ratio (SNR) values of the received signal, hypernetworks have been employed to generate the weight of a variable-node network in the Tanner graph [119]. Together, all the variable-node networks represent the graph neural network (GNN) on which message passing is performed. Meta-leaning algorithms have been explored in [111] as part of an end-to-end learning approach. There, meta-tasks were designed by varying the SNR to account for the task difficulty, under a convolutional encoder with a fixed coding rate of 1/2.

Overall, the aforementioned ML-based channel decoding methods can be classified into two categories [115]:

- *Data-driven methods:* These methods promote end-to-end learning approaches by substituting all the BP decoding components with a DNN [117]. Here, the structure of the code is ignored, and the channel decoding problem is regarded as a classification task from the input (i.e., received signal) to the output (i.e., decoded bits).
- *Model-driven methods:* the goal of this family of methods is to substitute the decoding components of the classical BP-based decoder (e.g., deinterleaver, log-likelihood ratio estimators) with trained DNNs without altering the classical sequence of decoding components [108], [120], [121].

Unlike the black-box use of DNNs for channel decoding, much less attention has been paid to studying how DG methodologies can be applied to both categories beyond the simple variation of the SNR values. Moreover, empirical and theoretical understanding of their potential for channel decoding is still lacking.

C. Channel Estimation

One of the crucial components of any wireless communication system is the channel estimator [122]. A vast body of prior work made use of data-driven ML techniques for channel estimation to show the attractive features of DNNs such as the low computational complexity at inference time [123], [124], [125], [126]. None of these studies, however, did analyze the impact of the distribution shifts on the reported estimation performance. Indeed, little effort has been devoted to investigating the robustness of DG algorithms in estimating wireless channels.

Another channel estimation algorithm for wideband mmWave systems was proposed in [127] based on unfolding the iterative shrinkage thresholding algorithm with a few learnable parameters. This algorithm was further extended to include a hypernetwork for the sake of generalization to new environments. Given the SNR level and the number of resolvable paths, the hypernetwork generates suitable learnable parameters for the channel estimation model. Alternatively, in [128], the authors proposed to train a hypernetwork to learn weighting factors so as to aggregate channel estimation models

Authorized licensed use limited to: Northeastern University. Downloaded on February 26,2024 at 14:58:14 UTC from IEEE Xplore. Restrictions apply.

learned for three main scenarios: urban micro, urban macro, and suburban macro. Hypernetwork recurrent DNNs have also been used to track wireless channels over a wide range of Doppler values [129]. For this multi-Doppler case, classical tracking methods make use of a bank of Kalman filters with an additional Doppler estimation step. Meta-learning was also adopted to train an encoder-decoder architecture to quickly adapt to new channel conditions by varying the number of pilot blocks preceding the payload in each transmission block [130]. For sparse MIMO channel estimation, the optimization/estimation modules of the approximate message passing (AMP) [131] and vector AMP (VAMP) [132] algorithms were substituted by learnable DNNs [133]. Specifically, DNNs did not neglect the "Onsager correction", which lies at the heart of the AMP paradigm and was rather employed to construct the underlying DNNs. By doing so, it was shown that the Onsager correction is beneficial to train DNNs that *i)* require fewer layers to reach a predefined level of accuracy and *ii)* yield greater accuracy overall as compared to DNNs ignoring the Onsager correction term.

Designing multiple channel estimation tasks pertaining to distinct domains requires varying wireless transmission parameters to simulate different channel communication scenarios. As depicted in Fig. 6, these parameters are categorized as:

- *Propagation parameters* which capture the different types of randomness in channel models [122]. They are not under control in practical communication scenarios.
- *System parameters* which govern multiple aspects of communication systems that are set by system designers such as the code rate, the number of transmit and receive antennas, the type and order of the modulation constellation, and the carrier frequency, etc.

It is worth noting that varying these parameters to generate different domains will lead to one or multiple types of distribution shifts. As one example, the design of a channel estimator for broadband communication has to generalize over the channel distributions. With the widely adopted strategy for bandwidth expansion, known as carrier aggregation [134], the distribution of the channel coefficient shifts across multiple non-contiguous narrow frequency bands. For this reason, assuming that the channel is the output of a DNN, the DNN-based channel estimator has to account for the label shift of the estimated channel coefficients because their support changes as a function of the frequency band.

Other related studies focusing on continual learning (CL) benchmarked the performance of CL-based methods for MIMO channels estimation by varying the SNR and the coherence time of the channel [135]. A continual learning minimum mean-square error (CL-MMSE) method has also been proposed in [136] where the DNN adapts to different numbers of receive antennas between 8 and 128 to generate tasks with different difficulties.

D. Beamforming

Steering the main lobe of antenna array systems toward users in a real-time manner (i.e., beamforming) is a critical task to minimize interference and enhance the achievable rate of wireless communication systems. This is because the antenna array processing in adaptive/reconfigurable digital signal processing algorithms assumes no mismatch between the actual and expected array responses to the received signal [137]. With the increase of the number of antenna elements in massive MIMO systems, a larger number of degrees of freedom is achieved at the cost of higher algorithmic complexity incurred when optimizing the beamformer weights [122]. Since beamforming weights must be continuously computed under changing propagation environments, ML methods have been explored as a possible solution to low-complexity beamforming design [138], [139]. For instance, the weighted minimum mean-square error (WMMSE) estimator of the transmit MISO beamforming vector was unfolded such that each estimation iteration corresponds to a DNN [140]. By doing so, the matrix-inverse operation of the standard WMMSE estimator is avoided in addition to the advantage of a lower computational complexity without sacrificing the estimation performance. It was also reported that fully distributed reinforcement learning (RL) estimates the uplink beamforming matrix by dividing the beamforming computations among distributed access points without significant accuracy deterioration [141]. We refer the reader to [142] for a comprehensive review of ML-based beamforming methods.

Few studies, however, have considered DG as an important ingredient to assess the performance of ML-aided beamforming solutions based on the meta-learning framework reviewed in Section III-C1. A meta-learning algorithm for weighted sum rate maximization was proposed for beamforming optimization in MISO downlink channels [143]. Instead of using the WMMSE algorithm iteratively to update each variable involved in the beamforming optimization problem, long-short-term-memory (LSTM) networks were used in the inner-loop of the meta-learning framework to learn the dynamic optimization strategy and hence update the optimization variables iteratively. The outer-loop of the meta-learning framework, however, makes use of the updated parameters to maximize the weighted sum rate. This strategy adaptively optimizes each variable with respect to the geometry of the sum-rate objective function, thereby achieving a better performance than the WMMSE algorithm. Another line of work employed the standard meta-learning MAML algorithm [87] for adaptive beamforming to new wireless environments [144]. This work was further extended to reduce the complexity of the MAML algorithm by dedicating a DNN model as a transferable feature extractor for feature reuse across wireless channel realizations [145]. Self-supervised learning was used to map uplink sub-6 GHz channels into mmWave beamforming vectors without accessing labeled training datasets [146]. By exploiting a dataset containing pairs of uplink and downlink channels, DNNs learned implicitly and autonomously the data representations from correlations in the training data pairs to predict the beamforming vectors.

E. Data Detection

To decrease the computational complexity of classical data detection algorithms, ML techniques were proposed to

Authorized licensed use limited to: Northeastern University. Downloaded on February 26,2024 at 14:58:14 UTC from IEEE Xplore. Restrictions apply.

detect communication signals under various conditions by reformulating bit/symbol detection as a conventional classification problem [147], [148], [149], [150]. In this context, the various DG techniques reviewed in Sections III-A–III-C can be leveraged to investigate the generalization capabilities of DNNs when applied to the data detection problem. For instance, DG demodulation methods for *multiple* modulation schemes have to account for both concept and label shifts of the estimated symbols because the modulation constellation varies from one domain to another. This scenario corresponds to wireless transmissions with adaptive modulation and coding where the choice of modulation order and coding rate is based on the instantaneous channel quality indicator (CQI).

Recently, data detection in MIMO systems with spatially correlated channels has been extensively studied. Indeed, MMNet [151] proposed an unfolding algorithm based on approximate message passing augmented by learnable parameters to achieve state-of-the-art performance on correlated channels. However, this algorithm needs to be re-trained for each channel realization. To overcome this drawback, the authors proposed to use a hypernetwork to predict the learnable parameters based on perfect CSI and noise power knowledge [152]. The generalization of this framework was tested under different SNR levels and user mobility settings to simulate different channel spatial correlations. One drawback of this approach is that it assumes that the CSI and noise power are perfectly known at the receiver. Similarly, the unfolded version of the expectation propagation detector was proposed wherein damping factors are learned using meta-learning [153]. This detector was also extended using hypernetworks to achieve generalization to new channel realizations and noise levels but for typical values of many other system parameters [154]. The major drawback here is that DNN must be retrained for each set of new system parameters. A meta-learning strategy was also used to train the damping factors of the VAMP algorithm to improve its convergence speed and quickly adapt to new environments, thereby yielding more accurate signal detection performance [153].

Other similar types of detection/recognition tasks are also of the same classification nature such as modulation classification in non-cooperative communication systems [155] and wireless transmitter classification [156]. These works focus on improving the classification accuracy only, and the generalization ability of DNNs was studied in a few prior works only [157].

In the rest of this section, we showcase an example from [158] using DG learning paradigms to enhance the accuracy of MIMO data detectors. We contrast in Fig. 9 receivers leveraging the structure of classical receivers in Fig. 9(a) and black-box end-to-end receivers depicted in Fig. 9(b). There, the DNN-aided receiver keeps using the classical demodulation and channel decoding modules. However, the symbol estimator module is a DNN-based demapper trained using both self-supervised learning and meta-learning:

- *Self-supervised learning:* The self-supervised training module is a DNN with parameters $\boldsymbol{\phi}$ and uses the training data from the channel encoder and modulator components. These two modules re-encode the successfully decoded bits with confident decisions to approximately reconstruct the transmitted symbols. For each information block, when the decoding is successful, the channel output $\boldsymbol{y}$ and the re-encoded symbol $\widetilde{\boldsymbol{s}}$ are used as data pair to update the symbol estimator parameters $\boldsymbol{\phi}$ for the next block. While this training procedure "recycles" the well-reconstructed symbols to train the self-supervised module without the knowledge of the transmitted symbols (c.f. Section III-C2), it does assume that channel conditions vary smoothly across blocks. This is because the symbol estimator trained on data from the tth block is assumed to correctly detect data from the $(t+1)$th channel realization. Since this assumption is inappropriate for tracking time-varying channels, it is critical to adapt the learning of the symbol estimator module to account for fast-varying channels.
- *Meta-learning:* To track time-varying channels, the parameters $\boldsymbol{\theta}$ of a meta-learner is optimized via the two-step optimization procedure of meta-learning (c.f. Section III-C1). This will enable fast and efficient adaptation of the parameters $\boldsymbol{\phi}$ of the symbol estimator. Meta-training is performed once every N blocks with P meta-iterations.

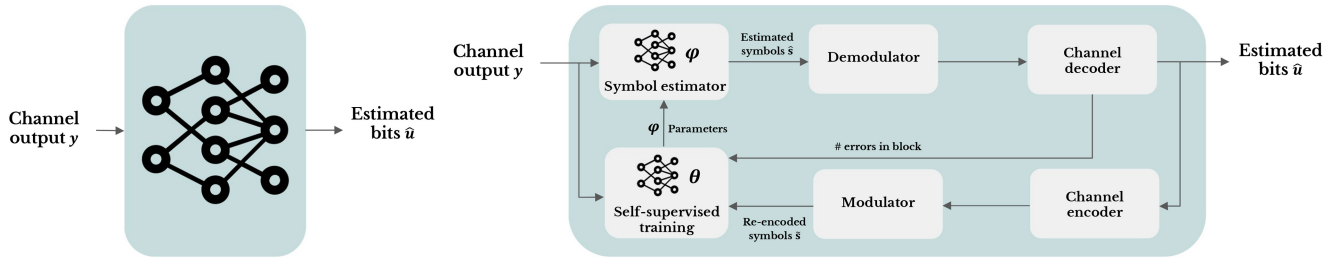
A key advantage of this DNN-aided decoder design is that it does not require the transmission of additional pilots during the test phase because the data is generated in a self-supervised manner. Moreover, the meta-training uses data batches corresponding to different channel realizations and hence does not assume temporal pattern similarity across past time blocks.

F. Beam Prediction

Since 6G and beyond communication systems are moving to higher frequency bands (e.g., mmWave and sub-terahertz), developing techniques for narrow directive beam management is critical to guarantee sufficient receive power. Existing solutions rely on leveraging the channel sparsity [159], constructing adaptive beam codebooks [160], and beam tracking [161]. Due to beam training overheads, these classical strategies, however, cannot meet the ever-increasing data rate demands of emerging applications for future systems with large antenna arrays serving highly-mobile users and latency-critical devices [162]. For these reasons, the development of ML-aided methods can offer data-driven solutions for the beam management problem because the beam direction decision depends on the user location and the geometry of the surroundings about which sensory datasets can be collected.

A practical ML solution is expected to generalize to unseen scenarios and operate in realistic dense deployments. The fact that practical sensors do not normally provide accurate enough positions/orientations for narrow beam alignment motivates acquiring multi-modality datasets about the environment such as sub-6GHz channel information, LiDAR point clouds, and radar measurements [163]. DG algorithms should be developed to leverage these datasets representing different domains in the same environment. For example, ideas from domain-invariant representations are beneficial to cope with distribution shift sources such as the quality of collected measurements (e.g., noise level, sensitivity to weather conditions),

Authorized licensed use limited to: Northeastern University. Downloaded on February 26,2024 at 14:58:14 UTC from IEEE Xplore. Restrictions apply.



(a) End-to-end receiver. (b) DNN-aided receiver.

Fig. 9. Two types of data-driven receivers for data detection: (a) an end-to-end receiver ignoring the structure of classical receivers, (b) a DNN-aided receiver exploiting the structure of classical modules using DG learning paradigms.

user mobility, and signal blockages. These factors lead to the acquisition of multiple data domains which can be exploited to learn both domain-invariant and domain-specific features to determine the index of the optimal beamforming vector from the codebook in a generalizable manner.

G. RIS-Aided Wireless Communications

Wireless communication aided by RISs has triggered a remarkable research effort in the last few years [164]. The possibility to purposely manipulate the electromagnetic propagation environment via the use of IRSs has pushed researchers to revisit fundamental wireless communication problems (e.g., beamforming, channel estimation) and incorporate the impact of RISs on the overall communication system performance measured in terms of capacity, estimation accuracy, secrecy, outage, and energy efficiency. In this context, ML methods belonging to multiple learning paradigms (e.g., supervised/unsupervised learning, reinforcement learning, federated learning) have been also devised to account for the propagation effects of IRSs. We refer the reader to the survey in [165] for an exhaustive summary of ML approaches for RIS-aided communication.

In regard to DG, only a handful of studies have assessed the performance of ML methods from the perspective of accuracy-generalization tradeoff. The problem of channel estimation for RIS-aided communication has been investigated in [166] where an adaptive shrinkage parameter based on a hypernetwork was used instead of a fixed shrinkage parameter. Based on the current channel recovery status, the hypernetwork provides an updated shrinkage parameter thanks to which the IRS-aided channel estimation accuracy has been assessed over different iterations as well as SNR values ranging between -10 dB and 25 dB. This work does not study DG as a function of the wireless communication parameters but rather with respect to the algorithmic steps of the LAMP algorithm. The robustness to additional noise of RL algorithms, when the CSI is perturbed, has been examined in [167] in the context of the optimization of RIS phase shifts. This work showed that RL methods exhibit resilience to different channel impairments as compared to classical optimization methods in the evaluation step only. In other words, DG training methodologies were not adopted and hence the work does not consider handling the domain shifts in estimating the phase shifts and only reports the performance degradation during inference.

H. Applications in Edge Networks

The domain shift problem arises naturally in IoT applications due to the heterogeneity in devices' behavior, spatial and temporal information, etc. For healthcare IoT sensors, the work in [168] applied a data alignment algorithm to learn and project accelerometer data from different users into a common feature space. The learned shared feature space is then used to track users' symptoms. For vehicle-to-everything (V2X) applications, a meta-learning approach for power allocation tasks has been proposed in [169] to enhance DNNs to achieve fast adaption to new environments with limited interactions.

DL has been applied in human activity recognition to extract meaningful features from raw sensory data instead of hand-engineered ones. Human activity recognition usually involves multi-modal sensory data from multiple devices/subjects to predict one or multiple activity labels. For the same activity, sensor data can vary depending on the subjects' characteristics such as gender, age, and behavior. One solution to this intra-activity shift problem is to remove the user-specific feature from the sensory information and keep the common activity features across all users only. To do so, feature disentanglement is proposed to learn two groups of representations: the common activity features and user-specific representations [170]. Another line of work focused on learning statistical features from sensory data using kernel-based techniques [171], [172]. These studies, however, make use of kernel-based methods for more predictive feature extraction from raw sensory data only. The use of kernel-based methods to improve DG as explained in Section III-B was not explored.

I. Summary and Lessons Learned

In the preceding sections, we reviewed different key applications in wireless communication where DG algorithms should be further investigated for the sake of robust generalization. Our observations and lessons learned are summarized below.

- *Lack of DG algorithms for wireless:* There is a small number of papers investigating DG in wireless communication compared to those applying deep learning papers for wireless communication in a black-box manner. This is to be opposed to the ML community where each top ML conference dedicates tracks and workshops for DG. Furthermore, DG papers from the ML community are continuously pinpointing to the fundamental limitations of DNNs to generalize in different contexts. The wireless

Authorized licensed use limited to: Northeastern University. Downloaded on February 26,2024 at 14:58:14 UTC from IEEE Xplore. Restrictions apply.

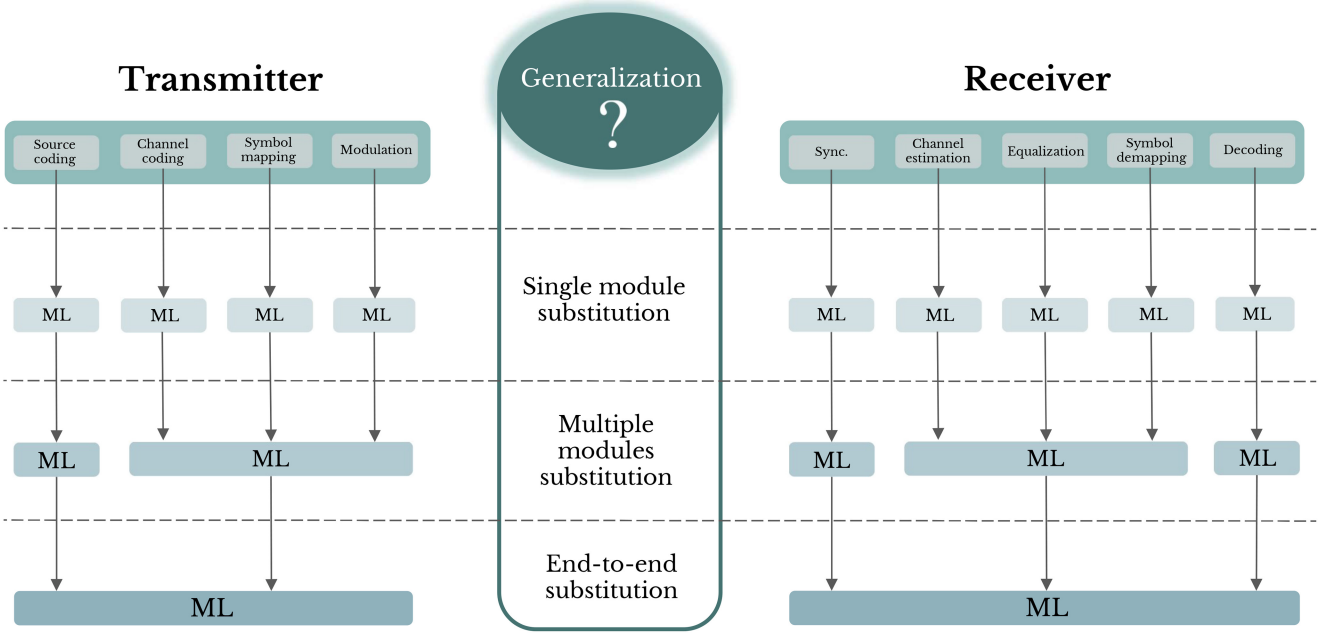


Fig. 10. The possible integration steps of ML methods into the conventional transmit/receive communication chain if ML methods will be proven to be robust to domain shifts.

communications researchers will need to focus on signal-based DG for wireless communications instead of just applying what the ML community proposes to solve some aspects of DG. To better judge the suitability of data-driven ML methods for real-world communication use cases, it is crucial to determine the uncertainty of ML algorithms and analyze their ability to generalize in order to lay the ground for rigorous evaluation protocols. However, minimal effort has been dedicated by the communication community to initiate such an investigation. As one example, use cases in the 3GPP Release 18 package such as CSI compression with autoencoders raise multiple interesting DG questions. Questions about the autoencoder training procedure, as well as, the different user traffic scenarios and urban areas should be considered before determining the source and target datasets.

- *One-sided focus on end-to-end DG:* Most DG algorithms are produced by the ML community and hence lack wireless communication knowledge in their designs. As a consequence, most DG communication papers make use of ML end-to-end techniques that are blind to the characteristics of the communication problem at hand. While this trend is worthwhile to assess the generalization performance of end-to-end learning methods, tailoring existing DG algorithms and devising new ones are essential research avenues that require further investigation.
- *Need for wireless DG benchmarks:* The number of DG benchmarks within the ML community has significantly increased over the last few years due to the need for algorithmic generalization evaluation (see [14] for a comprehensive review). Unfortunately, with few exceptions [111], [163], the absence of a unified benchmarking in wireless communications renders the comparison of the different proposed DG algorithms impossible. Consequently, it is crucial to establish a unified framework to analyze the improvements of the research endeavors and henceforth design robust and efficient DG algorithms.

V. Open Issues

In this section, we discuss some of the open questions that evolve around the necessity to carefully incorporate the DG concept in ML-oriented communication research. This is because, unlike many other ML-based technologies, most real-world communication applications require real-time operation and seamless adaptation to dynamically changing propagation conditions. This precludes the luxury of repeatedly training ML-oriented models and makes DG-induced robustness a must-have feature in any ML-aided communication system.

A. Beyond End-to-End Learning for Generalization

Most of the existing studies rely on end-to-end learning to train a holistic over-parametrized DNN architecture by applying gradient-based optimization to the learning system as a whole. This means that all transmit/receive modules of the communication system must be differentiable (in the reverse-mode algorithmic differentiation sense [173]). Few wireless communication libraries have been proposed to study differentiable communication systems [174], [175].

Before advocating the adequacy of applying ML methods to the building blocks of the wireless physical layer depicted in Fig. 10, DG has to be meticulously investigated and guaranteed *within* and *across* the blocks. From this perspective, it is not enough to claim the migration from model-based classical signal processing techniques to data-driven ML techniques without analyzing the impact of each migration on the overall system performance in terms of both accuracy and robustness.

Authorized licensed use limited to: Northeastern University. Downloaded on February 26,2024 at 14:58:14 UTC from IEEE Xplore. Restrictions apply.

While such migration is a conceptually profound paradigm shift, its impact continues to be assessed from the accuracy perspective only, and hence must also be carefully analyzed through the lens of generalization/robustness.

The legacy physical layer design strategy relies on the divide-and-conquer approach by decomposing (a.k.a. layering) the entire communication chain into smaller blocks [107]. Designing ML methods to substitute a single block or multiple blocks (see Fig. 10) raises critical generalization questions justified by the following two facts:

- End-to-end learning methods are trained with gradient descent-like optimizers, which exhibit slow convergence on ill-conditioned problems or convergence to possibly poor local optima. In other words, training is performed while hoping that the structural preconditioning is sufficiently strong to steer a method as simple as gradient descent from a random initial state to a highly non-trivial solution [176]. This assumption is risky since all ML techniques tailored for wireless applications are exclusively used for non-convex optimization problems.
- The valuable wireless communication know-how developed since the 50s is completely neglected during end-to-end training. "Standing on the shoulders of giants" (as Sir Isaac Newton once said) is a scientific tradition which promotes building upon the accumulated knowledge and discoveries made by others, and "end-to-end learning" must be proven robust to domain shifts to be considered an exception.

For these considerations, going beyond conventional end-to-end learning is an important step towards answering critical DG questions in data-driven ML techniques applied to wireless communications. In what follows, we discuss research directions to cope with some end-to-end learning limitations.

B. Hybrid Data-Driven and Model-Driven Methods

After more than a century-long research effort in radio communications, state-of-the-art communication modeling, and fast estimation algorithms are becoming more essential to high-bandwidth transmissions. From a DG perspective, the power of these classical model-driven tools lies in their guaranteed generalization capabilities because they do not depend on specific domains that are tied to generated/collected datasets. This generalization, however, often comes at the cost of high complexity.

Data-driven methods can come into play as an effective tool to reduce the computational complexity of classical model-based methods at the cost of generalization. As advocated in [177], a hybrid framework that combines the benefits of both data-driven and model-based techniques is worth pursuing. Adopting this framework will prevent the generated domains for DG from being fully dependent on *i)* the convergence of gradient-based optimizers for data-driven methods, or *ii)* the complexity of model-based methods. Moreover, exploiting the structure of the model allows DG learning paradigms to be applied to a subset of the DNN parameters, thereby facilitating their scaling. For example, the use case we described in Section IV-E showed how restricting the use of DNNs for symbol demapping only while keeping the other classical modules can be beneficial for meta-learning to generalize to both slow and fast varying channel realizations. For better illustration, we elaborate in what follows on how data-driven methods can be combined with physically consistent model-based methods.

The study of DG for MIMO communication should benefit from the side information provided by the physical laws governing the wave transmission and the circuits of RF components (i.e., amplifiers, and antennas). By employing physically consistent models [41], [178], [179], it is possible to exploit the inherent symmetries and invariances in communication scenarios owing to Maxwell's equations [48], [180]. From this perspective, physically consistent models for wireless communications offer an opportunity to generate communication datasets that exhibit domain-invariant regularities (e.g., antenna impedances), thereby diminishing the generalization difficulties across domains. As one example, fixing the impedance matrices of transmit and receive linear/planar antenna arrays increases the amount of correlation in the wireless channel, which can be exploited by DNNs for better channel estimation accuracy.

Moreover, this physically consistent direction opens the door for the analysis of DG through the lens of antenna theory. For example, it might be possible to determine which spacing parameter of the antenna array provides the best DNN accuracy for channel estimation. By doing so, realistic wireless communication domains are generated and more faithful representations of the real-world transmissions are simulated, thereby leading to a physically consistent version of digital twins for wireless communications [181].

C. From Image-Based DG Methods to Signal-Based Methods

Existing DG methodologies have been predominantly geared towards image-based vision tasks, leaving signal-based tasks almost unexplored despite being versatile in several real-world applications such as healthcare, retail, climate, finance, and communication. This unbalanced exploration impacts the development of specific DG methods for signal-based tasks. For instance, feature alignment approaches for DG are relying heavily on DNNs as feature extractors which are specifically fine-tuned to vision tasks, thereby leaving DG feature extraction for non-image signals severely underexplored. Some work looked at temporal distributional shifts in clinical healthcare [182], [183] and climate [184] applications, but none of the prior work explored it in wireless communication.

From this perspective, we highlight the importance of taking the first step towards a deeper understanding of temporal distributional shifts in wireless communication due to dynamic changes in the received signal resulting from the varying propagation properties (e.g., coherence time and Doppler shift).

D. Compound Domain Generalization

As mentioned previously, most of the presented methods for DG assume a homogeneous setting where domain labels are available. However, this assumption may not be realistic

Authorized licensed use limited to: Northeastern University. Downloaded on February 26,2024 at 14:58:14 UTC from IEEE Xplore. Restrictions apply.

in several problems where the domain labels are hard to obtain or define. In this case, several techniques discussed above either become inapplicable (e.g., meta-learning) or their performance degrades drastically [7]. Recently, there has been a surge of interest in studying the compound DG setting in vision problems. Most of the methods for compound DG propose to infer latent domain information from data and then use standard learning techniques to generalize across the latent domains. These solutions are, however, based on different restrictive assumptions such as *i)* the latent domains are distinct and separable [103], *ii)* the domain heterogeneity originates from stylistic differences [185] or *iii)* the latent domains are balanced [186]. Compound DG is hence still an active research field with a lot of room for improvement, especially in wireless communication problems.

E. Federated Domain Generalization

Distributed learning algorithms enable devices to cooperatively build a unified learning model across agents with local training. As a result, a wide variety of distributed ML methods have been proposed and extensively analyzed within the federated learning (FL) framework [187].

For wireless physical layer applications, FL has been explored to address multiple key communication problems beyond the data security aspect [188] such as channel estimation [189], symbol detection [190] and beamforming [191]. All of these works do not assume the availability of a central entity (e.g., base station) at which the learning model is trained. However, the question of whether the model learned by each agent generalizes to unseen scenarios is still unanswered and this remains an unexplored research area. In the context of IoT applications, very few efforts started investigating the challenges of DG for IoT devices by aligning each device's domain to a reference distribution in a distributed manner [192].

Addressing DG in the FL context is known as *federated domain generalization* (FDG) [193]. Distributed agents can collect their local data independently, hence naturally forming a distinct source domain. At the time of writing, no research paper in wireless communication has studied FDG, e.g., in the context of distributed MIMO [194] consisting of distributed antenna array systems.

VI. Conclusion

Studying the impact of distribution shifts on the performance of ML-based algorithms for wireless applications is of paramount importance to our research community to better reflect on the adequacy of adopting the data-driven ML approaches in communication systems engineering. In particular, the investigation of domain generalization will lay the ground for rigorous evaluation protocols of data-driven algorithms for wireless communications systems. In this paper, we presented an overview of state-of-the-art methodologies for domain generalization problems to handle distribution shifts. To justify the need to devise new algorithms with better generalization capabilities, we distinguished the four types of distribution shifts between source and target domains. We also provided an overview of multiple important fields related to generalization to better put domain generalization in proper perspective across close research areas. Then, we summarized the three existing methodologies to improve the generalization capabilities of deep learning models, namely, data manipulation, data representation, and domain generalization learning paradigms. In doing so, we gave multiple examples and suggestions not covered in the current literature where these methodologies can be applied to wireless communication applications. We then reviewed the recent research contributions to improve the generalization of neural network models when solving wireless communication problems. These problems involve beam prediction, data detection, channel decoding, beamforming, edge networks, etc. We also presented the learned lessons from the existing applications of domain generalization methodologies for wireless communication problems by highlighting the lack of *i)* algorithms exploiting the domain knowledge from well-established communication models, and *ii)* open-source benchmarks to accelerate the development of robust algorithms for future wireless networks. Finally, we discussed open questions to enrich and bridge the gap between both domain generalization and wireless communication fields.

References

[1] 3GPP release 18. "Study on artificial intelligence (AI)/machine learning (ML) for NR air interface RAN." Mar. 2023. [Online]. Available: https://www.3gpp.org/ftp/tsg_ran/WG1_RL1/TSGR1_112/Report

[2] M. Belgiovine, K. Sankhe, C. Bocanegra, D. Roy, and K. R. Chowdhury, "Deep learning at the edge for channel estimation in beyond-5G massive MIMO," *IEEE Wireless Commun.*, vol. 28, no. 2, pp. 19–25, Apr. 2021.

[3] X. Liu et al., "Domain generalization under conditional and label shifts via variational Bayesian inference," in *Proc. 13th Int. Joint Conf. Artif. Intell. (IJCAI)*, Aug. 2021, pp. 881–887.

[4] D. Hendrycks and T. G. Dietterich, "Benchmarking neural network robustness to common corruptions and perturbations," 2019, *arxiv.abs/1903.12261*.

[5] F. Zhuang et al., "A comprehensive survey on transfer learning," *Proc. IEEE*, vol. 109, no. 1, pp. 43–76, Jan. 2021.

[6] M. Wang and W. Deng, "Deep visual domain adaptation: A survey," *Neurocomputing*, vol. 312, pp. 135–153, Oct. 2018.

[7] K. Zhou, Z. Liu, Y. Qiao, T. Xiang, and C. C. Loy, "Domain generalization: A survey," 2021, *arxiv.abs/2103.02503*.

[8] D. Li, Y. Yang, Y. Song, and T. M. Hospedales, "Deeper, broader and artier domain generalization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5543–5551.

[9] Y. Li, Y. Yang, W. Zhou, and T. M. Hospedales, "Feature-critic networks for heterogeneous domain generalization," in *Proc. 36th Int. Conf. Mach. Learn. (ICML)*, 2019, pp. 3915–3924.

[10] J. Miller, K. Krauth, B. Recht, and L. Schmidt, "The effect of natural distribution shift on question answering models," in *Proc. 37th Int. Conf. Mach. Learn. (ICML)*, 2020, pp. 6905–6916.

[11] N. Joshi and H. He, "An investigation of the (in)effectiveness of counterfactually augmented data," in *Proc. 60th Annu. Meeting Assoc. Comput. Linguist.*, 2022, pp. 3668–3681.

[12] C. Li et al., "Domain generalization on medical imaging classification using episodic training with task augmentation," *Comput. Biol. Med.*, vol. 141, Feb. 2022, Art. no. 105144.

[13] S. H. Silva and P. Najafirad, "Opportunities and challenges in deep learning adversarial robustness: A survey," 2020, *arxiv.abs/2007.00753*.

[14] J. Wang, C. Lan, C. Liu, Y. Ouyang, and T. Qin, "Generalizing to unseen domains: A survey on domain generalization," in *Proc. 13th Int. Joint Conf. Artif. Intell. (IJCAI)*, 2021, pp. 4627–4635.

[15] P. Sheth, R. Moraffah, K. S. Candan, A. Raglin, and H. Liu, "Domain generalization—A causal perspective," 2022, *arXiv:2209.15177*.

[16] I. Gulrajani and D. Lopez-Paz, "In search of lost domain generalization," in *Proc. 9th Int. Conf. Learn. Rep. (ICLR)*, May 2021, p. 5.

Authorized licensed use limited to: Northeastern University. Downloaded on February 26,2024 at 14:58:14 UTC from IEEE Xplore. Restrictions apply.

[17] P. W. Koh et al., "WILDS: A benchmark of in-the-wild distribution shifts," in *Proc. 38th Int. Conf. Mach. Learn.*, Jul. 2021, pp. 5637–5664.
[18] J.-C. Gagnon-Audet, K. Ahuja, M.-J. Darvishi-Bayazi, P. Mousavi, G. Dumas, and I. Rish, "WOODs: Benchmarks for out-of-distribution generalization in time series," 2022, *arXiv:2203.09978*.
[19] F. R. Kschischang, B. J. Frey, and H. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.
[20] H. Shimodaira, "Improving predictive inference under covariate shift by weighting the log-likelihood function," *J. Stat. Plan. Inference*, vol. 90, no. 2, pp. 227–244, 2000.
[21] W. M. Kouw, "An introduction to domain adaptation and transfer learning," 2018, *arxiv.abs/1812.11806*.
[22] R. Caruana, "Multitask learning," *Mach. Learn.*, vol. 28, no. 1, pp. 41–75, 1997.
[23] W. Wang, V. W. Zheng, H. Yu, and C. Miao, "A survey of zero-shot learning: Settings, methods, and applications," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, pp. 1–13, 2019.
[24] Y. Sun, X. Wang, Z. Liu, J. Miller, A. A. Efros, and M. Hardt, "Test-time training for out-of-distribution generalization," 2019, *arxiv.abs/1909.13231*.
[25] K. Peng, Z. Wu, and J. Ernst, "Zero-shot deep domain adaptation," in *Proc. 15th Eur. Conf. Comput. Vis. (ECCV)*, Munich, Germany, Sep. 2018, pp. 793–810.
[26] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *Proc. 2nd Int. Conf. Learn. Rep. (ICLR)*, 2014, p. 6.
[27] I. J. Goodfellow et al., "Generative adversarial networks," 2014, *arxiv.abs/1406.2661*.
[28] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. J. Smola, "A kernel two-sample test," *J. Mach. Learn. Res.*, vol. 13, no. 1, pp. 723–773, 2012.
[29] I. O. Tolstikhin, O. Bousquet, S. Gelly, and B. Schölkopf, "Wasserstein auto-encoders," 2017, *arxiv.abs/1711.01558*.
[30] K. Zhou, Y. Yang, T. M. Hospedales, and T. Xiang, "Learning to generate novel domains for domain generalization," in *Proc. 16th Eur. Conf. Comput. Vis. (ECCV)*, Glasgow, U.K., Aug. 2020, pp. 561–578.
[31] L. Li et al., "Progressive domain expansion network for single domain generalization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2021, pp. 224–233.
[32] H. Zhang, M. Cissé, Y. N. Dauphin, and D. Lopez-Paz, "MIXUP: Beyond empirical risk minimization," in *Proc. 6th Int. Conf. Learn. Rep. (ICLR)*, May 2018, pp. 1–9.
[33] W. Wang, S. Liao, F. Zhao, C. Kang, and L. Shao, "DomainMix: Learning generalizable person re-identification without human annotations," in *Proc. 32nd Brit. Mach. Vis. Conf. (BMVC)*, Nov. 2021, p. 355.
[34] Y. Wang, H. Li, and A. C. Kot, "Heterogeneous domain generalization via domain mixup," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, May 2020, pp. 3622–3626.
[35] Y. Shu, Z. Cao, C. Wang, J. Wang, and M. Long, "Open domain generalization with domain-augmented meta-learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2021, pp. 9624–9633.
[36] K. Zhou, Y. Yang, Y. Qiao, and T. Xiang, "Domain generalization with mixstyle," in *Proc. 9th Int. Conf. Learn. Rep. (ICLR)*, 2021, p. 6.
[37] F. Qiao and X. Peng, "Uncertainty-guided model generalization to unseen domains," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 6790–6800.
[38] Q. Xu, R. Zhang, Y. Zhang, Y. Wang, and Q. Tian, "A fourier-based framework for domain generalization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 14383–14392.
[39] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *J. Big Data*, vol. 6, p. 60, Jul. 2019.
[40] T. Raviv and N. Shlezinger, "Data augmentation for deep receivers," *IEEE Trans. Wireless Commun.*, early access, Mar. 30, 2023, doi: 10.1109/TWC.2023.3261782.
[41] M. T. Ivrlac and J. A. Nossek, "Toward a circuit theory of communication," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 57, no. 7, pp. 1663–1683, Jul. 2010.
[42] M. Akrout, V. Shyianov, F. Bellili, A. Mezghani, and R. W. Heath, "Achievable rate of near-field communications based on physically consistent models," *IEEE Trans. Wireless Commun.*, vol. 22, no. 2, pp. 1266–1280, Feb. 2023.
[43] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 23–30.
[44] G. Pialla, M. Devanne, J. Weber, L. Idoumghar, and G. Forestier, "Data augmentation for time series classification with deep learning models," in *Proc. Adv. Anal. Learn. Temp. Data (AALTD)*, 2022, pp. 117–132.
[45] S. Shankar, V. Piratla, S. Chakrabarti, S. Chaudhuri, P. Jyothi, and S. Sarawagi, "Generalizing across domains via cross-gradient training," in *Proc. 6th Int. Conf. Learn. Rep. (ICLR)*, May 2018, p. 6.
[46] R. Volpi, H. Namkoong, O. Sener, J. C. Duchi, V. Murino, and S. Savarese, "Generalizing to unseen domains via adversarial data augmentation," in *Proc. Adv. Neural Inf. Process. Syst. Annu. Conf. Neural Inf. Process. Syst. (NeurIPS)*, Dec. 2018, pp. 5339–5349.
[47] K. Zhou, Y. Yang, T. M. Hospedales, and T. Xiang, "Deep domain-adversarial image generation for domain generalisation," in *Proc. 34th AAAI Conf. Artif. Intell. (AAAI)*, Feb. 2020, pp. 13025–13032.
[48] J. D. Jackson, *Classical Electrodynamics*. Hoboken, NJ, USA: Wiley, 1999.
[49] R. G. Hohlfeld and N. Cohen, "Self-similarity and the geometric requirements for frequency independence in antennae," *Fractals*, vol. 7, no. 1, pp. 79–84, 1999.
[50] Y. Bengio, A. C. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.
[51] A. V. Oppenheim, J. S. Lim, G. E. Kopec, and S. C. Pohlig, "Phase in speech and pictures," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Apr. 1979, pp. 632–637.
[52] A. V. Oppenheim and J. S. Lim, "The importance of phase in signals," *Proc. IEEE*, vol. 69, no. 5, pp. 529–541, May 1981.
[53] B. C. Hansen and R. F. Hess, "Structural sparseness and spatial phase alignment in natural scenes," *J. Opt. Soc. America A*, vol. 24, no. 7, pp. 1873–1885, 2007.
[54] W. Lu, J. Wang, H. Li, Y. Chen, and X. Xie, "Domain-invariant feature exploration for domain generalization," in *Proc. 30th Int. Conf. Mach. Learn.*, 2011, pp. 1–9.
[55] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.
[56] B. Schölkopf, A. Smola, and K.-R. Müller, "Kernel principal component analysis," in *Proc. Int. Conf. Artif. Neural Netw.*, 1997, pp. 583–588.
[57] G. Blanchard, A. A. Deshmukh, Ü. Dogan, G. Lee, and C. Scott, "Domain generalization by marginal transfer learning," *J. Mach. Learn. Res.*, vol. 22, p. 2, Jan. 2021.
[58] K. Muandet, D. Balduzzi, and B. Schölkopf, "Domain generalization via invariant feature representation," in *Proc. 30th Int. Conf. Mach. Learn. (ICML)*, Jun. 2013, pp. 10–18.
[59] S. M. Erfani et al., "Robust domain generalisation by enforcing distribution invariance," in *Proc. 25th Int. Joint Conf. Artif. Intell. (IJCAI)*, Jul. 2016, pp. 1455–1461.
[60] S. Hu, K. Zhang, Z. Chen, and L. Chan, "Domain generalization via multidomain discriminant analysis," in *Proc. 35th Conf. Uncertainty Artif. Intell. (UAI)*, Jul. 2019, pp. 292–302.
[61] M. Ghifary, D. Balduzzi, W. B. Kleijn, and M. Zhang, "Scatter component analysis: A unified framework for domain adaptation and domain generalization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 7, pp. 1414–1430, Jul. 2017.
[62] H. Li, S. J. Pan, S. Wang, and A. C. Kot, "Domain generalization with adversarial feature learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 5400–5409.
[63] R. Gong, W. Li, Y. Chen, and L. V. Gool, "DLOW: Domain flow for adaptation and generalization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2477–2486.
[64] Y. Li et al., "Deep domain generalization via conditional invariant adversarial networks," in *Proc. 15th Eur. Conf. Comput. Vis. (ECCV)*, Munich, Germany, Sep. 2018, pp. 647–663.
[65] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Deep domain confusion: Maximizing for domain invariance," 2014, *arxiv.abs/1412.3474*.
[66] F. Zhou, Z. Jiang, C. Shui, B. Wang, and B. Chaib-Draa, "Domain generalization via optimal transport with metric similarity learning," *Neurocomputing*, vol. 456, pp. 469–480, Oct. 2021.
[67] X. Peng, Q. Bai, X. Xia, Z. Huang, K. Saenko, and B. Wang, "Moment matching for multi-source domain adaptation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Seoul, South Korea, Oct./Nov. 2019, pp. 1406–1415.
[68] X. Pan, P. Luo, J. Shi, and X. Tang, "Two at once: Enhancing learning and generalization capacities via IBN-Net," in *Proc. 15th Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 484–500.

Authorized licensed use limited to: Northeastern University. Downloaded on February 26,2024 at 14:58:14 UTC from IEEE Xplore. Restrictions apply.

[69] H. Nam and H. Kim, "Batch-instance normalization for adaptively style-invariant neural networks," in *Proc. Adv. Neural Inf. Process. Syst. Annu. Conf. Neural Inf. Process. Syst. (NeurIPS)*, Dec. 2018, pp. 2563–2572.

[70] L. Qi, L. Wang, Y. Shi, and X. Geng, "Unsupervised domain generalization for person re-identification: A domain-specific adaptive framework," 2021, *arxiv.abs/2111.15077*.

[71] X. Fan, Q. Wang, J. Ke, F. Yang, B. Gong, and M. Zhou, "Adversarially adaptive normalization for single domain generalization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 8208–8217.

[72] M. Arjovsky, L. Bottou, I. Gulrajani, and D. Lopez-Paz, "Invariant risk minimization," 2019, *arxiv.abs/1907.02893*.

[73] Y. J. Choe, J. Ham, and K. Park, "An empirical study of invariant risk minimization," 2020, *arxiv.abs/2004.05007*.

[74] A. Sonar, V. Pacelli, and A. Majumdar, "Invariant policy optimization: Towards stronger generalization in reinforcement learning," in *Proc. 3rd Annu. Conf. Learn. Dyn. Control (L4DC)*, 2021, pp. 21–33.

[75] J. Mitrovic, B. McWilliams, J. C. Walker, L. H. Buesing, and C. Blundell, "Representation learning via invariant causal mechanisms," in *Proc. 9th Int. Conf. Learn. Rep. (ICLR)*, May 2021, pp. 1–9.

[76] D. Krueger et al., "Out-of-distribution generalization via risk extrapolation (REX)," in *Proc. 38th Int. Conf. Mach. Learn. (ICML)*, Jul. 2021, pp. 5815–5826.

[77] K. Ahuja et al., "Invariance principle meets information bottleneck for out-of-distribution generalization," in *Proc. Annu. Conf. Adv. Neural Inf. Process. Syst.*, Dec. 2021, pp. 3438–3450.

[78] A. Khosla, T. Zhou, T. Malisiewicz, A. A. Efros, and A. Torralba, "Undoing the damage of dataset bias," in *Proc. 12th Eur. Conf. Comput. Vis. Comput. Vis. (ECCV)*, Florence, Italy, Oct. 2012, pp. 158–171.

[79] L. Niu, W. Li, and D. Xu, "Multi-view domain generalization for visual recognition," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Santiago, Chile, Dec. 2015, pp. 4193–4201.

[80] Z. Ding and Y. Fu, "Deep domain generalization with structured low-rank constraint," *IEEE Trans. Image Process.*, vol. 27, no. 1, pp. 304–313, Jan. 2018.

[81] A. Zunino et al., "Explainable deep classification models for domain generalization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPR Workshops)*, Jun. 2021, pp. 3233–3242.

[82] Y. Wang, H. Li, L. Chau, and A. C. Kot, "Variational disentanglement for domain generalization," 2021, *arxiv.abs/2109.05826*.

[83] M. Ilse, J. M. Tomczak, C. Louizos, and M. Welling, "DIVA: Domain invariant variational autoencoders," in *Proc. Int. Conf. Med. Imag. Deep Learn. (MIDL)*, 2020, pp. 322–348.

[84] F. Qiao, L. Zhao, and X. Peng, "Learning to learn single domain generalization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2020, pp. 12553–12562.

[85] H. Zhang, Y. Zhang, W. Liu, A. Weller, B. Schölkopf, and E. P. Xing, "Towards principled disentanglement for domain generalization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 8014–8024.

[86] M. Huisman, J. N. van Rijn, and A. Plaat, "A survey of deep meta-learning," *Artif. Intell. Rev.*, vol. 54, no. 6, pp. 4483–4541, 2021.

[87] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proc. 34th Int. Conf. Mach. Learn. (ICML)*, Aug. 2017, pp. 1126–1135.

[88] T. M. Hospedales, A. Antoniou, P. Micaelli, and A. J. Storkey, "Meta-learning in neural networks: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 9, pp. 5149–5169, Sep. 2022.

[89] D. Li, Y. Yang, Y. Song, and T. M. Hospedales, "Learning to generalize: Meta-learning for domain generalization," in *Proc. 32nd AAAI Conf. Artif. Intell. (AAAI) 30th Innov. Appl. Artif. Intell. (IAAI) 8th AAAI Symp. Educ. Adv. Artif. Intell. (EAAI)*, Feb. 2018, pp. 3490–3497.

[90] Y. Li and N. Vasconcelos, "REPAIR: removing representation bias by dataset resampling," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9572–9581.

[91] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," 2021, *arXiv:2111.06377*.

[92] D. Kim, Y. Yoo, S. Park, J. Kim, and J. Lee, "SelFreg: Self-supervised contrastive regularization for domain generalization," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9599–9608.

[93] Z.-H. Zhou, *Ensemble Methods: Foundations and Algorithms*. Hoboken, NJ, USA: Wiley, 2012.

[94] K. Zhou, Y. Yang, Y. Qiao, and T. Xiang, "Domain adaptive ensemble learning," *IEEE Trans. Image Process.*, vol. 30, pp. 8008–8018, 2021.

[95] A. D'Innocente and B. Caputo, "Domain generalization with domain-specific aggregation modules," in *Proc. 40th German Conf. Pattern Recognit. (GCPR)*, Oct. 2018, pp. 187–198.

[96] M. Mancini, S. R. Bulò, B. Caputo, and E. Ricci, "Best sources forward: Domain generalization through source-specific nets," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Athens, Greece, Oct. 2018, pp. 1353–1357.

[97] S. Seo, Y. Suh, D. Kim, G. Kim, J. Han, and B. Han, "Learning to optimize domain specific normalization for domain generalization," in *Proc. 16th Eur. Conf. Comput. Vis. (ECCV)*, Glasgow, U.K., Aug. 2020 pp. 68–83.

[98] M. Segù, A. Tonioni, and F. Tombari, "Batch normalization embeddings for deep domain generalization," *Pattern Recognit.*, vol. 135, Mar. 2023, Art. no. 109115.

[99] P. Izmailov, D. Podoprikhin, T. Garipov, D. P. Vetrov, and A. G. Wilson, "Averaging weights leads to wider optima and better generalization," in *Proc. 34th Conf. Uncertainty Artif. Intell. (UAI)*, Aug. 2018, pp. 876–885.

[100] J. Cha et al., "SWAD: Domain generalization by seeking flat minima," in *Proc. Adv. Neural Inf. Process. Syst. Annu. Conf. Neural Inf. Process. Syst. (NeurIPS)*, Dec. 2021, pp. 22405–22418.

[101] D. Ha, A. M. Dai, and Q. V. Le, "Hypernetworks," 2016, *arxiv.abs/1609.09106*.

[102] T. Volk, E. Ben-David, O. Amosy, G. Chechik, and R. Reichart, "Example-based hypernetworks for out-of-distribution generalization," 2022, *arxiv.abs/2203.14276*.

[103] J. Qu, T. Faney, Z. Wang, P. Gallinari, S. Yousef, and J. de Hemptinne, "HMOE: hypernetwork-based mixture of experts for domain generalization," 2022, *arxiv.abs/2211.08253*.

[104] G. Parascandolo, A. Neitz, A. Orvieto, L. Gresele, and B. Schölkopf, "Learning explanations that are hard to vary," 2020, *arXiv:2009.00329*.

[105] M. Koyama and S. Yamaguchi, "When is invariance useful in an out-of-distribution generalization problem?" 2020, *arXiv:2008.01883*.

[106] S. Hemati, G. Zhang, A. Estiri, and X. Chen, "Understanding hessian alignment for domain generalization," 2023, *arXiv:2308.11778*.

[107] R. G. Gallager, *Principles of Digital Communication*, vol. 1. Cambridge, U.K.: Cambridge Univ. Press, 2008.

[108] Y. Jiang, H. Kim, H. Asnani, S. Kannan, S. Oh, and P. Viswanath, "Turbo autoencoder: Deep learning based channel codes for point-to-point communication channels," in *Proc. Adv. Neural Inf. Process. Syst. Annu. Conf. Neural Inf. Process. Syst. (NeurIPS)*, vol. 32, Dec. 2019, pp. 2754–2764.

[109] H. Kim, Y. Jiang, S. Kannan, S. Oh, and P. Viswanath, "DeepCode: Feedback codes via deep learning," *IEEE J. Sel. Areas Inf. Theory*, vol. 1, no. 1, pp. 194–206, May 2020.

[110] M. Kim, R. Fritschek, and R. F. Schaefer, "Learning end-to-end channel coding with diffusion models," in *Proc. 26th Int. ITG Workshop Smart Antennas 13th Conf. Syst. Commun. Coding (WSA SCC)*, 2023, pp. 1–6.

[111] R. Li et al., "A channel coding benchmark for meta-learning," in *Proc. Neural Inf. Process. Syst. Datasets Benchmarks (NeurIPS)*, Dec. 2021, pp. 1–9.

[112] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: Turbo-codes," *IEEE Trans. Commun.*, vol. 44, no. 10, pp. 1261–1271, Oct. 1996.

[113] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 619–637, Feb. 2001.

[114] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. New York, NY, USA: Morgan Kaufmann, 1988.

[115] K. Niu, J. Dai, K. Tan, and J. Gao, "Deep learning methods for channel decoding: A brief tutorial," in *Proc. 10th IEEE/CIC Int. Conf. Commun. China (ICCC)*, Xiamen, China, Jul. 2021, pp. 144–149.

[116] P. Trifonov, "Efficient design and decoding of polar codes," *IEEE Trans. Commun.*, vol. 60, no. 11, pp. 3221–3227, Nov. 2012.

[117] T. Gruber, S. Cammerer, J. Hoydis, and S. T. Brink, "On deep learning-based channel decoding," in *Proc. 51st Annu. Conf. Inf. Sci. Syst. (CISS)*, Baltimore, MD, USA, Mar. 2017, pp. 1–6.

[118] E. Nachmani, Y. Be'ery, and D. Burshtein, "Learning to decode linear codes using deep learning," in *Proc. IEEE 54th Annu. Allerton Conf. Commun. Control Comput. (Allerton)*, Sep. 2016, pp. 341–346.

[119] E. Nachmani and L. Wolf, "Hyper-graph-network decoders for block codes," in *Proc. Adv. Neural Inf. Process. Syst. Annu. Conf. Neural Inf. Process. Syst. (NeurIPS)*, Dec. 2019, pp. 2326–2336.

[120] E. Nachmani, E. Marciano, L. Lugosch, W. J. Gross, D. Burshtein, and Y. Be'ery, "Deep learning methods for improved decoding of linear codes," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 119–131, Feb. 2018.

Authorized licensed use limited to: Northeastern University. Downloaded on February 26,2024 at 14:58:14 UTC from IEEE Xplore. Restrictions apply.

[121] B. Vasic, X. Xiao, and S. Lin, "Learning to decode LDPC codes with finite-alphabet message passing," in *Proc. Inf. Theory Appl. Workshop (ITA)*, San Diego, CA, USA, Feb. 2018, pp. 1–9.

[122] R. W. Heath and A. Lozano, *Foundations of MIMO Communication*. Cambridge, U.K.: Cambridge Univ. Press, 2018.

[123] H. Ye, G. Y. Li, and B. Juang, "Power of deep learning for channel estimation and signal detection in OFDM systems," *IEEE Wireless Commun. Lett.*, vol. 7, no. 1, pp. 114–117, Feb. 2018.

[124] Q. Hu, F. Gao, H. Zhang, S. Jin, and G. Y. Li, "Deep learning for channel estimation: Interpretation, performance, and comparison," *IEEE Trans. Wireless Commun.*, vol. 20, no. 4, pp. 2398–2412, Apr. 2021.

[125] M. Soltani, V. Pourahmadi, A. Mirzaei, and H. Sheikhzadeh, "Deep learning-based channel estimation," *IEEE Commun. Lett.*, vol. 23, no. 4, pp. 652–655, Apr. 2019.

[126] A. K. Gizzini, M. Chafii, A. Nimr, and G. Fettweis, "Deep learning based channel estimation schemes for IEEE 802.11p standard," *IEEE Access*, vol. 8, pp. 113751–113765, 2020.

[127] W. Jin, H. He, C.-K. Wen, S. Jin, and G. Y. Li, "Adaptive channel estimation based on model-driven deep learning for wideband mmWave systems," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, 2021, pp. 1–6.

[128] P. Jiang, C.-K. Wen, S. Jin, and G. Y. Li, "Dual CNN-based channel estimation for MIMO-OFDM systems," *IEEE Trans. Commun.*, vol. 69, no. 9, hboxpp. 5859–5872, Sep. 2021.

[129] K. Pratik, R. A. Amjad, A. Behboodi, J. B. Soriaga, and M. Welling, "Neural augmentation of Kalman filter with hypernetwork for channel tracking," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Madrid, Spain, Dec. 2021, pp. 1–6.

[130] S. Park, O. Simeone, and J. Kang, "End-to-end fast training of communication links without a channel model via online meta-learning," in *Proc. 21st IEEE Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Atlanta, GA, USA, May 2020, pp. 1–5.

[131] D. L. Donoho, A. Maleki, and A. Montanari. "How to design message passing algorithms for compressed sensing." 2011. [Online]. Available: https://arxiv.org/abs/0907.3574

[132] S. Rangan, P. Schniter, and A. K. Fletcher, "Vector approximate message passing," *IEEE Trans. Inf. Theory*, vol. 65, no. 10, pp. 6664–6684, Nov. 2019.

[133] M. Borgerding, P. Schniter, and S. Rangan, "AMP-inspired deep networks for sparse linear inverse problems," *IEEE Trans. Signal Process.*, vol. 65, no. 16, pp. 4293–4308, Aug. 2017.

[134] G. Yuan, X. Zhang, W. Wang, and Y. Yang, "Carrier aggregation for LTE-advanced mobile communication systems," *IEEE Commun. Mag.*, vol. 48, no. 2, pp. 88–93, Feb. 2010.

[135] M. Akrout, A. Feriani, F. Bellili, A. Mezghani, and E. Hossain, "Continual learning-based MIMO channel estimation: A benchmarking study," 2022, *arXiv:2211.10753*.

[136] S. Kumar, S. K. Vankayala, B. S. Sahoo, and S. Yoon, "Continual learning-based channel estimation for 5G millimeter-wave systems," in *Proc. IEEE 18th Annu. Consum. Commun. Netw. Conf. (CCNC)*, 2021, pp. 1–6.

[137] R. A. Monzingo and T. W. Miller, *Introduction to Adaptive Arrays*. New York, NY, USA: Scitech, 2004.

[138] Z. D. Zaharis et al., "Implementation of antenna array beamforming by using a novel neural network structure," in *Proc. Int. Conf. Telecommun. Multimedia (TEMU)*, Jul. 2016, pp. 1–5.

[139] Z. D. Zaharis, I. P. Gravas, P. I. Lazaridis, T. V. Yioultsis, C. S. Antonopoulos, and T. D. Xenos, "An effective modification of conventional beamforming methods suitable for realistic linear antenna arrays," *IEEE Trans. Antennas Propag.*, vol. 68, no. 7, pp. 5269–5279, Jul. 2020.

[140] L. Pellaco, M. Bengtsson, and J. Jaldén, "Matrix-inverse-free deep unfolding of the weighted MMSE beamforming algorithm," *IEEE Open J. Commun. Soc.*, vol. 3, pp. 65–81, 2022.

[141] F. Fredj, Y. F. Al-Eryani, S. Maghsudi, M. Akrout, and E. Hossain, "Distributed beamforming techniques for cell-free wireless networks using deep reinforcement learning," *IEEE Trans. Cogn. Commun. Netw.*, vol. 8, no. 2, pp. 1186–1201, Jun. 2022.

[142] H. A. Kassir et al., "A review of the state of the art and future challenges of deep learning-based beamforming," *IEEE Access*, vol. 10, pp. 80869–80882, 2022.

[143] J. Xia and D. Gündüz, "Meta-learning based beamforming design for MISO downlink," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2021, pp. 2954–2959.

[144] Y. Yuan, G. Zheng, K. Wong, B. E. Ottersten, and Z. Luo, "Transfer learning and meta learning-based fast downlink beamforming adaptation," *IEEE Trans. Wireless Commun.*, vol. 20, no. 3, pp. 1742–1755, Mar. 2021.

[145] J. Zhang, Y. Yuan, G. Zheng, I. Krikidis, and K. Wong, "Embedding model-based fast meta learning for downlink beamforming adaptation," *IEEE Trans. Wireless Commun.*, vol. 21, no. 1, pp. 149–162, Jan. 2022.

[146] I. Chafaa, R. Negrel, E. V. Belmega, and M. Debbah, "Self-supervised deep learning for mmWave beam steering exploiting sub-6 GHz channels," *IEEE Trans. Wireless Commun.*, vol. 21, no. 10, pp. 8803–8816, Oct. 2022.

[147] S. Dörner, S. Cammerer, J. Hoydis, and S. ten Brink, "Deep learning based communication over the air," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 132–143, Feb. 2018.

[148] N. Farsad and A. J. Goldsmith, "Detection algorithms for communication systems using deep learning," 2017, *arxiv.abs/1705.08044*.

[149] N. Samuel, T. Diskin, and A. Wiesel, "Deep MIMO detection," in *Proc. 18th IEEE Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Sapporo, Japan, Jul. 2017, pp. 1–5.

[150] A. Al-Baidhani and H. H. Fan, "Learning for detection: A deep learning wireless communication receiver over Rayleigh fading channels," in *Proc. IEEE Int. Conf. Comput. Netw. Commun. (ICNC)*, 2019, pp. 6–10.

[151] M. K. Shirkoohi, M. Alizadeh, J. Hoydis, and P. Fleming, "Adaptive neural signal detection for massive MIMO," *IEEE Trans. Wireless Commun.*, vol. 19, no. 8, pp. 5635–5648, Aug. 2020.

[152] M. Goutay, F. A. Aoudia, and J. Hoydis, "Deep hypernetwork-based MIMO detection," in *Proc. 21st IEEE Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, May 2020, pp. 1–5.

[153] J. Zhang, Y. He, Y.-W. Li, C.-K. Wen, and S. Jin, "Meta learning-based MIMO detectors: Design, simulation, and experimental test," *IEEE Trans. Wireless Commun.*, vol. 20, no. 2, pp. 1122–1137, Feb. 2021.

[154] J. Zhang, C.-K. Wen, and S. Jin, "Adaptive MIMO detector based on hypernetwork: Design, simulation, and experimental test," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 1, pp. 65–81, Jan. 2022.

[155] T. J. O'Shea, J. Corgan, and T. C. Clancy, "Convolutional radio modulation recognition networks," in *Proc. Int. Conf. Eng. Appl. Neural Netw.*, 2016, pp. 213–226.

[156] K. Youssef, L. Bouchard, K. Haigh, J. Silovsky, B. Thapa, and C. Vander Valk, "Machine learning approach to RF transmitter identification," *IEEE J. Radio Freq. Identification*, vol. 2, no. 4, pp. 197–205, Dec. 2018.

[157] J. Liu, T. Oyedare, and J. Park, "Detecting out-of-distribution data in wireless communications applications of deep learning," *IEEE Trans. Wireless Commun.*, vol. 21, no. 4, pp. 2476–2487, Apr. 2022.

[158] T. Raviv, S. Park, O. Simeone, Y. C. Eldar, and N. Shlezinger, "Online meta-learning for hybrid model-based deep receivers," *IEEE Trans. Wireless Commun.*, vol. 22, no. 10, pp. 6415–6431, Oct. 2023.

[159] R. W. Heath, N. G. Prelcic, S. Rangan, W. Roh, and A. M. Sayeed, "An overview of signal processing techniques for millimeter wave MIMO systems," *IEEE J. Sel. Topics Signal Process.*, vol. 10, no. 3, pp. 436–453, Apr. 2016.

[160] M. Alrabeiah, Y. Zhang, and A. Alkhateeb, "Neural networks based beam codebooks: Learning mmWave massive MIMO beams that adapt to deployment and hardware," *IEEE Trans. Commun.*, vol. 70, no. 6, pp. 3818–3833, Jun. 2022.

[161] S. Jayaprakasam, X. Ma, J. W. Choi, and S. Kim, "Robust beam-tracking for mmWave mobile communications," *IEEE Commun. Lett.*, vol. 21, no. 12, pp. 2654–2657, Dec. 2017.

[162] K. Ma, Z. Wang, W. Tian, S. Chen, and L. Hanzo, "Deep learning for beam management: Opportunities, state-of-the-arts and challenges," 2021, *arxiv.abs/2111.11177*.

[163] G. Charan, U. Demirhan, J. Morais, A. Behboodi, H. Pezeshki, and A. Alkhateeb, "Multi-modal beam prediction challenge 2022: Towards generalization," 2022, *arxiv.abs/2209.07519*.

[164] S. Hu, F. Rusek, and O. Edfors, "Beyond massive MIMO: The potential of data transmission with large intelligent surfaces," *IEEE Trans. Signal Process.*, vol. 66, no. 10, pp. 2746–2758, May 2018.

[165] K. M. Faisal and W. Choi, "Machine learning approaches for reconfigurable intelligent surfaces: A survey," *IEEE Access*, vol. 10, pp. 27343–27367, 2022.

[166] W. Tsai, C. Chen, C. Teng, and A. Wu, "Low-complexity compressive channel estimation for IRS-aided mmWave systems with hypernetwork-assisted LAMP network," *IEEE Commun. Lett.*, vol. 26, no. 8, pp. 1883–1887, Aug. 2022.

[167] A. Feriani, A. Mezghani, and E. Hossain, "On the robustness of deep reinforcement learning in irs-aided wireless communications systems," 2021, *arxiv.abs/2107.08293*.

Authorized licensed use limited to: Northeastern University. Downloaded on February 26,2024 at 14:58:14 UTC from IEEE Xplore. Restrictions apply.

[168] A. Feriani, A. Refaey, and E. Hossain, "Tracking pandemics: A MEC-enabled IoT ecosystem with learning capability," *IEEE Internet Things Mag.*, vol. 3, no. 3, pp. 40–45, Sep. 2020.
[169] Y. Yuan, G. Zheng, K. Wong, and K. B. Letaief, "Meta-reinforcement learning based resource allocation for dynamic V2X communications," *IEEE Trans. Veh. Technol.*, vol. 70, no. 9, pp. 8964–8977, Sep. 2021.
[170] J. Su, Z. Wen, T. Lin, and Y. Guan, "Learning disentangled behaviour patterns for wearable-based human activity recognition," *Proc. ACM Interact. Mobile Wearable Ubiquitous Technol.*, vol. 6, no. 1, pp. 1–19, 2022.
[171] H. Qian, S. J. Pan, and C. Miao, "Sensor-based activity recognition via learning from distributions," in *Proc. 32nd AAAI Conf. Artif. Intell. (AAAI) 30th Innov. Appl. Artif. Intell. (IAAI) 8th AAAI Symp. Educ. Adv. Artif. Intell. (EAAI)*, Feb. 2018, pp. 6262–6269.
[172] H. Qian, S. J. Pan, B. Da, and C. Miao, "A novel distribution-embedded neural network for sensor-based activity recognition," in *Proc. 28th Int. Joint Conf. Artif. Intell. (IJCAI)*, 2019, pp. 5614–5620.
[173] B. Speelpenning, *Compiling Fast Partial Derivatives of Functions Given by Algorithms*, Univ. Illinois at Urbana–Champaign, Champaign, IL, USA, 1980.
[174] J. Hoydis et al., "SIONNA: An open-source library for next-generation physical layer research," Mar. 2022. [Online]. Available: https://arxiv.org/abs/2203.11854
[175] "AI for wireless with MATLAB." Accessed: Feb. 1, 2023. [Online]. Available: https://www.mathworks.com/solutions/wireless-communications/ai.html
[176] T. Glasmachers, "Limits of end-to-end learning," in *Proc. 9th Asian Conf. Mach. Learn. (ACML)*, Seoul, South Korea, Nov. 2017, pp. 17–32.
[177] L. Pellaco, "Machine learning for wireless communications: Hybrid data-driven and model-based approaches," Ph.D. dissertation, Dept. Comput. Sci., KTH Royal Inst. Technol., Stockholm, Sweden, 2022.
[178] M. Akrout et al., "Super-wideband massive MIMO," 2022, *arxiv.abs/2208.01556*.
[179] A. Pizzo, L. Sanguinetti, and T. L. Marzetta, "Fourier plane-wave series expansion for holographic MIMO communications," *IEEE Trans. Wireless Commun.*, vol. 21, no. 9, pp. 6890–6905, Sep. 2022.
[180] C. E. Baum and H. N. Kritikos, *Symmetry in Electromagnetics*, vol. 1. London, U.K.: Taylor & Francis, 1985, pp. 1–90.
[181] L. U. Khan, W. Saad, D. Niyato, Z. Han, and C. S. Hong, "Digital-twin-enabled 6G: Vision, architectural trends, and future directions," *IEEE Commun. Mag.*, vol. 60, no. 1, pp. 74–80, Jan. 2022.
[182] L. L. Guo et al., "Evaluation of domain generalization and adaptation on improving model robustness to temporal dataset shift in clinical medicine," *Sci. Rep.*, vol. 12, no. 1, pp. 1–10, 2022.
[183] B. Ma, H. Li, W. Zheng, and B. Lu, "Reducing the subject variability of EEG signals with adversarial domain generalization," in *Proc. Neural Inf. Process. 26th Int. Conf. (ICONIP)*, Dec. 2019, pp. 30–42.
[184] A. Malinin et al., "SHIFTs: A dataset of real distributional shift across multiple large-scale tasks," in *Proc. Neural Inf. Process. Syst. Datasets Benchmarks (NeurIPS)*, Dec. 2021, p. 5.
[185] C. Chen, J. Li, X. Han, X. Liu, and Y. Yu, "Compound domain generalization via meta-knowledge encoding," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, New Orleans, LA, USA, Jun. 2022, pp. 7109–7119.
[186] T. Matsuura and T. Harada, "Domain generalization using a mixture of multiple latent domains," 2019, *arxiv.abs/1911.07661*.
[187] J. Konečný, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," 2016, *arxiv.abs/1610.05492*.
[188] H. Zhang, C. Yang, and B. Dai, "When wireless federated learning meets physical layer security: The fundamental limits," in *Proc. IEEE INFOCOM Conf. Comput. Commun. Workshops (INFOCOM)*, May 2022, pp. 1–6.
[189] A. M. Elbir and S. Coleri, "Federated learning for channel estimation in conventional and RIS-assisted massive MIMO," *IEEE Trans. Wireless Commun.*, vol. 21, no. 6, pp. 4255–4268, Jun. 2022.
[190] M. B. Mashhadi, N. Shlezinger, Y. C. Eldar, and D. Gündüz, "Fedrec: Federated learning of universal receivers over fading channels," in *Proc. IEEE Stat. Signal Process. Workshop (SSP)*, Jul. 2021, pp. 576–580.
[191] A. M. Elbir and S. Coleri, "Federated learning for hybrid beamforming in mm-Wave massive MIMO," *IEEE Commun. Lett.*, vol. 24, no. 12, pp. 2795–2799, Dec. 2020.
[192] L. Zhang, X. Lei, Y. Shi, H. Huang, and C. Chen, "Federated learning for IoT devices with domain generalization," *IEEE Internet Things J.*, vol. 10, no. 11, pp. 9622–9633, Jun. 2023.
[193] L. Zhang, X. Lei, Y. Shi, H. Huang, and C. Chen, "Federated learning with domain generalization," 2021, *arxiv.abs/2111.10487*.
[194] D. Wang, J. Wang, X. You, Y. Wang, M. Chen, and X. Hou, "Spectral efficiency of distributed MIMO systems," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 10, pp. 2112–2127, Oct. 2013.

Mohamed Akrout received the B.E. degree in computer engineering from the Ecole Polytechnique de Montreal, Montreal, Canada, and the "Diplome d'Ingenieur" degree from Telecom ParisTech (ENST), Paris, France, both in 2016, and the M.S. degree in artificial intelligence from the University of Toronto, Toronto, Canada, in 2018. He is currently pursuing the Ph.D. degree with the Electrical and Computer Engineering Department, University of Manitoba, Winnipeg, Canada. He has published in topics related to statistical inference, wireless communication, computational neuroscience, and deep learning. His current research interests include domain generalization for deep learning models, signal processing for inverse problems, and physically consistent antenna design. He was the recipient of the Doctoral Scholarship from the Natural Sciences and Engineering Research Council of Canada in 2022.

Amal Feriani received the B.E. degree in engineering from the Ecole Polytechnique de Tunisie in 2014, the M.S. degree in data science from the University of Paris Dauphine, Paris, France, in 2016, and the M.Sc. degree in machine learning and wireless communication from the University of Manitoba, Winnipeg, Canada, in 2021, where she is currently working as a Senior Machine Learning Engineer. Her research interests include multiagent systems, wireless communication, and large language models.

Faouzi Bellili (Member, IEEE) received the Diplome d'Ingénieur degree from Tunisia Polytechnic School in 2007, and the M.Sc. and Ph.D. degrees (with highest Hons.) from the Institut National de la Recherche Scientifique, University of Quebec, Montreal, QC, Canada, in 2009 and 2014, respectively.

He is currently an Assistant Professor with the Department of Electrical and Computer Engineering, University of Manitoba, Winnipeg, MB, Canada. From September 2014 to September 2016, he was working as a Research Associate with INRS-EMT and from December 2016 to May 2018, he was a Postdoctoral Fellow with the ECE Department, University of Toronto, Toronto, ON, Canada. His research focuses on statistical and array signal processing for wireless communications. He was awarded the very prestigious NSERC PDF Grant from 2017 to 2018. He was also awarded another prestigious PDF Scholarship offered over the same period (but declined) from the "Fonds de Recherche du Quebec Nature et Technologies". He received the INRS Innovation Award in 2014 and 2015, the very prestigious Academic Gold Medal of the Governor General of Canada in 2009 and 2010, and the Excellence Grant of the Director General of INRS in 2009 and 2010. He received the Award of the best M.Sc. Thesis in INRS-EMT in 2009 and 2010 and twice—for both the M.Sc. and Ph.D. programs—the National Grant of Excellence from the Tunisian Government. In 2011, he was also awarded the Merit Scholarship for Foreign Students from the Ministere de l'Education, du Loisir et du Sport of Quebec, Canada.

Authorized licensed use limited to: Northeastern University. Downloaded on February 26,2024 at 14:58:14 UTC from IEEE Xplore. Restrictions apply.

Amine Mezghani (Member, IEEE) received the Ph.D. degree in electrical engineering from the Technical University of Munich, Germany, in 2015. He is currently an Assistant Professor with the Department of Electrical and Computer Engineering, University of Manitoba, Canada. Prior to this, he was a Postdoctoral Fellow with the University of Texas at Austin, USA, and a Postdoctoral Scholar with the Department of Electrical Engineering and Computer Science, University of California at Irvine, Irvine, USA. He has published more than a hundred papers, particularly on the topic of signal processing and communications with low-resolution analog-to-digital and digital-to-analog converters. His current research interests include millimeter-wave communications, massive MIMO, hardware constrained communication systems, antenna theory, and large-scale signal processing algorithms. He was the recipient of 2021 IEEE Signal Processing Society Best Paper Award, the 2023 Winnipeg Rh Institute Foundation Award for outstanding research accomplishments, and the 2016 joint Rohde & Schwarz and EE Department Outstanding Dissertation Award.

Ekram Hossain (Fellow, IEEE) is a Professor and an Associate Head (Graduate Studies) with the Department of Electrical and Computer Engineering, University of Manitoba, Canada (https://home.cc.umanitoba.ca/hossaina/). He is a member (Class of 2016) of the College of the Royal Society of Canada, and a Fellow of the Canadian Academy of Engineering and the Engineering Institute of Canada. His current research interests include design, analysis, and optimization of wireless networks with emphasis on beyond 5G cellular networks. He received the 2017 IEEE ComSoc TCGCC (Technical Committee on Green Communications and Computing) Distinguished Technical Achievement Recognition Award "for outstanding technical leadership and achievement in green wireless communications and networking". He was listed as a Clarivate Analytics Highly Cited Researcher in Computer Science from 2017 to 2022. He served as the Editor-in-Chief for the IEEE PRESS from 2018 to 2021 and the IEEE COMMUNICATIONS SURVEYS AND TUTORIALS from 2012 to 2016. He was an Elected Member of the Board of Governors of the IEEE ComSoc from 2018 to 2020, and served as the Director for Magazines of IEEE ComSoc from 2020 to 2021. He served as the Director for Online Content of IEEE ComSoc from 2022 to 2023. He was elevated to an IEEE Fellow "for contributions to spectrum management and resource allocation in cognitive and cellular radio networks."

Authorized licensed use limited to: Northeastern University. Downloaded on February 26,2024 at 14:58:14 UTC from IEEE Xplore. Restrictions apply.