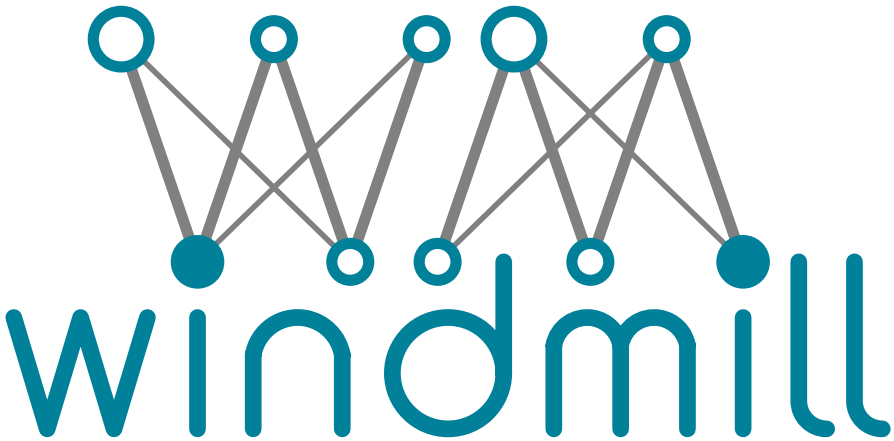
Marie Skłodowska Curie Action

WINDMILL

Machine Learning for Wireless Communications

H2020-MSCA-ITN-ETN

Grant Agreement Number: 813999



WP3–Advancing the field of ML for wireless communications

D3.1–Identify challenges of conventional ML in wireless networks

Contractual Delivery Date:	August 1, 2020
Actual Delivery Date:	August 4, 2020
Responsible Beneficiary:	ETHZ
Contributing Beneficiaries:	ETHZ, Eurecom, CTTC
Dissemination Level:	Public
Version:	1.0

PROPRIETARY RIGHTS STATEMENT
This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 813999.

Document Information

Document ID:	WP3/D3.1
Version Date:	August 4, 2020
Total Number of Pages:	37
Abstract:	Machine learning and information theory are highly interconnected because they both turn statistical relations between data into actionable solutions. While information theory mainly deals with communications problems and have clear models to do so, machine learning mostly relies on collected data. In this document, we review open issues in both machine learning and communications and how they are intertwined. We especially devote a chapter to transfer learning, which would be paramount for learning from data in communications networks. We also dedicate another chapter to GANs and how they can be used to learn communications simulators from data. Finally, we revisit the need for machine learning for MIMO.
Keywords:	Machine Learning, Kernel methods, Generative Adversarial Networks, MIMO

Authors

Full name	**Beneficiary/ Organisation**	**e-mail**	**Role**
Shirin Goshtasbpour	ETH Zurich	shirin.goshtasbpour@inf.ethz.ch	Contributor
Roberto Pereira	CTTC	rpereira@cttc.es	Contributor
Davit Gogolashvili	Eurecom	davit.gogolashvili@eurecom.fr	Contributor
Fernando Perez-Cruz	ETH Zurich	fernando.perezcruz@sdsc.ethz.ch	Coordinator

Table of Contents

List of Figures

1. Introduction

Machine Learning and Information/Communication Theory are two sides of the same coin. While the cornerstone of information theory is, the channel capacity formula [1]

$$C = \max_{p_X(x)} I(X;Y) \quad \text{bits per channel use,} \tag{1.1}$$

which turns a statistical concept, i.e. the mutual information between two random variables X and Y, into an actionable concept, i.e. the maximum information that can be transmitted through a communication channel when $p_{Y|X}(y|x)$ is fixed and known. Most of the work when designing communications system goes towards achieving low complexity communication systems that achieve capacity[1].
Machine learning cornerstone classification formula [2] is given by

$$\max_{\theta} E_{p_X(x)} \left[\log q_\theta(x)\right], \tag{1.2}$$

Which measures the negative cross-entropy between the underlying process that generates our data, i.e. $p_X(x)$, and our model to describe this data, i.e. $q_\theta(x)$.
In both cases, statistical measures between random variables become operational in communications and classification. This is also the point in which they deviate. Information/-Communication Theory rely on good and simple models for the channel, e.g. $p_{Y|X}(y|x)$. This has allowed tremendous progress since the 1950s to our day. This can be seen very clearly on the changing nature of the modulation scheme in mobile communication networks from FDMA (1G) to TDMA (2G) to CDMA (3G) to OFDMA (4G/5G), as technology improves new theoretical solutions can be implemented to improve the efficiency of digital mobile networks. But similar conclusions could be drawn for modem design in phone lines or the improvements in fiber optics.
The progress in machine learning has been less linear with many highs and lows, in which many of the hypes in the sixties and eighties were followed by times in which machine learning (or artificial intelligence) was considered not viable for solving long standing problems in society. In retrospect it is easy to understand those failures, given that in machine learning we do not have simple models that represent our problems, but we rely on data. For example in Kevin Murphy's book [2], Machine learning is defined: "... as a set of methods that that can automatically detect patterns in data, ...". We did not have access to enough data until the mid-2000s. This is basically the difference between today's wave of machine learning and previous waves is that we have been collecting data for over 20 years and that that data is mostly well organized and amenable for being using by machine learning algorithms. The celebrated Alexnet [3], which changed the Imagnet competition in 2010, has almost the same structure than the LeNet from 1989 [4]. They just had access to enough data[2] to be able to make much better than human engineered features.
Much of the progress in machine learning has come from deep learning procedures, in which a representation of the data is also learnt together with the classifier/regressor. This avoids the feature engineering that was needed for classical models like Random Forests,

[1] This is self-evident in the Physical and MAC layers, but it permeates all the others

[2] Computational resources also plays a role, but it can be seen that the way Alexnet was trained using a personal computer and not a state-of-the-art computer cluster.

or Support Vector Machines. On the other hand, deep learning procedures need significantly more data to be able learn the features as well as the relation with data. Most of the progress in deep learning has come from disciplines, like image classification or natural language processing in which data has been collected and shared among research groups and companies that has helped with progress.
The improvements of machine learning in general and deep learning in particular has increased the visibility of machine learning for many other disciplines, in search of better statistical tools to solve long standing problems. Digital communication is one of those, in which machine learning and deep learning is having a significant impact. For example, there has been workshops and tutorials in Machine learning for communication the past three IEEE Globecoms [5–7] and there will be a Selected Areas in Communications for Machine Learning at 2021 IEEE ICC [8]. Also, there has been an effort to collect all the recent advances in deep learning for communications [9] in which some of the relevant papers have been collected and summarized. Most of these papers shows that machine learning can improve linear time algorithms that are currently in used, which can become available in mobile devises, because the computing power of them has increased significantly and there is significant progress in bringing NNs to low cost edge devices too.
Now, if machine learning would have a long-lasting impact in digital communications, we will need to rethink how do we design and implement wireless communications networks. Because in the late 1980s, there were some papers proposing NNs for channel equalization, e.g. [10], that were never realized in practice. First, most of the previous work, assumes that the channel models are still valid and that simulations are the way forward when showing new results in digital communications. The application of machine learning is mostly on the receiver end of the communication chain, where improved detection and regression algorithm drives the improved in performance. But this should be a gateway to collect data from life networks in which machine learning receivers can truly show their learning and adapting capabilities, especially if some of the assumptions for which the communications networks were built for do not (always) hold. Finally, this should spill over to how do we design the transmitter too. While in most machine learning application the role of the transmitter is left to nature, in the design of communication systems, transmitters are human design.
In the remaining chapters of this document we revisit several aspects open in machine learning and digital communications. For example in the second chapter, we illustrate the problem of transfer learning, when there has been a covariate shift in the data. Transfer learning would be extremely important tool, when we train receivers in real scenarios and how do we transfer them to other parts of the networks. In the third chapter, we revisit simulations of communications networks using ML. Instead of relying on simulations, we put forward the idea of taking data from communications networks and learn its behavior using a generative adversarial network. In the final chapter, we dive deep into the application of machine learning to Massive MIMO and what are the advantages it can bring to the field.

2. Learning With Random Features Under Covariate Shift

2.1. Introduction

Consider supervised learning problem which is in general the minimization of risk functional [11]

$$R[P, f] = E_P L(f(X), Y) = \int L(f(x), y) dP(x, y) \tag{2.1}$$

of a loss function L. The problem is to choose in the set of function F a function $f(x)$ which minimizes the risk when the probability distribution P is unknown but random independent observations $(x_1, y_1), \dots, (x_n, y_n)$ are given. Replacing P in (2.1) with an empirical measure $P_n = 1/n \sum_{i=1}^{n} \delta_{(x_i, y_i)}$ we get an empirical risk minimization problem and to avoid overfitting we add a regularization term $\Omega(\theta)$

$$R[P_n, f] = \frac{1}{n} \sum_{i=1}^{n} L(f(x_i), y_i) + \lambda \Omega(\theta). \tag{2.2}$$

2.2. Covariate shift

Situation described above corresponds to the scenario when training and test data are drown identically and independently from the same distribution P. The problem becomes more challenging when the training set is drawn from P_{tr} but our goal is to minimize risk with respect to the different test distribution $R[P_{te}, f]$. Importance sampling identity

$$E_{P_{te}} L(f(X), Y) = E_{P_{tr}} \frac{P_{te}(X, Y)}{P_{tr}(X, Y)} L(f(X), Y) \tag{2.3}$$

allows us to compute risk with respect to P_{te} using P_{tr}. Since our objective is to minimize the risk functional (2.3) based on the observations $(x_1, y_1), \dots, (x_n, y_n)$ from the training distribution P_{tr} empirical risk minimization (2.2) doesn't necessarily provide a good inference. Intuitively one can achieve better performance replacing P_{te} by weighted empirical measure $P_n^w = \frac{1}{n} \sum_{i=1}^{n} w(x_i, y_i) \delta_{(x_i, y_i)}$ with weights equal to $w(x_i, y_i) = P_{te}(x_i, y_i)/P_{tr}(x_i, y_i)$ which leads us to the weighted empirical risk minimization

$$R[P_n^w, f] = \frac{1}{n} \sum_{i=1}^{n} w(x_i, y_i) L(f(x_i), y_i) + \lambda \Omega(\theta). \tag{2.4}$$

We will assume that $P_{te,Y|X} = P_{tr,Y|X} = P_{Y|X}$ allowing $P_{tr,X}$ to differ from $P_{te,X}$. This particular case of dataset shift is known as a covariate shift [12]. Under this assumption for weights in (2.3), we have $w(x, y) = w(x) = P_{te}(x)/P_{tr}(x)$, so that weights only depends on the training and test distribution of the independent variable.
Covariate shift phenomenon occure in many real-world applications such as off-policy reinforcement learning [13], spam filtering [14] or brain-computer interfacing [15].

2.3. Random Feature Approximation

Kernel based learning algorithms such as Support Vector Machines [16] or Gaussian process regression [17] are very useful and popular learning tools mainly because their flexibility and ability to approximate any regression or classification function with enough training data. But the main drawback is the computation needed to manipulate with kernel matrix. For example kernel ridge regression requires $O(n^2)$ is space to store kernel K and roughly $O(n^3)$ to invert this matrix. Similar requirements are shared by other methods [16].
Embedding the input space into the high dimensional feature space $\psi : \mathcal{X} \to \mathcal{F}$ makes the original data linearly separable in $\mathcal{F}$ usually by increasing the dimension of the space. For a given embedding ψ we define the kernel function as an inner product in $\mathcal{F}$, $K(x, x') = \langle \psi(x), \psi(x') \rangle$. the basic idea of random feature [18] is to relax this equality assuming it holds only approximately

$$K(x, x') = \langle \psi(x), \psi(x') \rangle \approx \phi_M(x)\phi_M(x')$$

where feature function $\phi_M(x) = (\phi(x, \omega_1), \dots, \phi(x, \omega_M))$ parameterized by some random vector $w_i \in \Omega$. With the low-dimensional approximation of the kernel we reduce the computational cost as soon as M $\leq$ n. However, the price we pay for the computational efficiency is to decrease the complexity of the model. In fact random feature is a randomized finite dimensional approximation of potentially infinite dimensional feature map. There are few results considering generalization properties of of learning with random features [19], [20].
A question we are going to study here is to investigate the generalization properties of random feature under covariate shift.

2.4. Main Result

We will consider the setting similar to [19]. The largest space we will consider is

$$F_R = \left\{ \int \alpha(\omega)\phi(x, \omega)d\omega : |\alpha(\omega)| \leq Rp(\omega) \right\} \tag{2.5}$$

and the estimator is

$$\hat{f}(x) = \sum_{i=1}^{M} \phi(x, \omega_i)\hat{\alpha}_i, \quad \hat{\alpha} = \underset{\|\alpha\|_\infty \leq R}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} w(x_i) L \left(\sum_{j=1}^{M} \phi(x_i, \omega_j)\alpha_j, y_i \right) \tag{2.6}$$

where L is L-Lipschitz loss function with the property $L(y, y') = L(yy')$, and $\|\alpha\|_\infty = \max\{|\alpha_1|, \dots, |\alpha_M|\}$. Also we will require that $|\phi(x, w)| \leq 1$ for all $(x, w) \in \mathcal{X} \times \Omega$. We will denote by $\hat{F}$ the space of all solution of (2.6).
Our main result gives us an upper bound for the excess risk under covariate shift if $w(x)$ is chosen perfectly in the population sense [21]. Assume that $w(x) \in [0, W]$ and $P_{te} = w(x)P_{tr}$. Let the training data $\{(x_1, y_1), \dots, (x_n, y_n)\}$ drawn iid from the training distribution P_{tr}. Than for any $\delta > 0$, estimator (2.6) gives a function $\hat{f}(x)$ that satisfies

$$\begin{aligned} R[P_{te}, \hat{f}] - \min_{f \in F_M} R[P_{te}, f] &\leq 2LR/\sqrt{n} + LRW\sqrt{2\ln(2/\delta)/n} \\ &|L(0)|W\sqrt{2\ln(2/\delta)/n} + \frac{LRW}{\sqrt{M}} \left(1 + \sqrt{2\ln(1/\delta)}\right) \end{aligned} \tag{2.7}$$

with probability at least $1-2\delta$ over training distribution and the choice of the parameters $\omega_1, \ldots, \omega_M$.
The proof of the theorem goes in a standard way by bounding the estimation and approximation errors. For the approximation error we will use the Lemma 2 from [19].

Lemma 1(Approximation Error). *Assume that $w(x) \in [0, W]$, $P_{te} = w(x)P_{tr}$ and $L(y, y')$ is L-Lipschitz loss function in its first argument. Then for every $f \in F_R$ and any $\delta > 0$ there exists a function $\hat{f} \in \hat{F}$ such that with probability at least $1-\delta$ over $\omega_1, \ldots, \omega_M \sim P_\omega$*

$$R[P_{te}, \hat{f}] - R[P_{te}, f] \leq \frac{LRW}{\sqrt{M}} \left(1 + \sqrt{2\ln(1/\delta)}\right) \tag{2.8}$$

For the estimation error bound we will adopt the proof of the theorem 8 from [22] which in our case will be

Lemma 2(Estimation Error). *Assume that $w(x) \in [0, W]$, $P_{te} = w(x)P_{tr}$ and $L(y, y') = L(yy')$ is L-Lipschitz loss function. Let the training data $\{(x_i, y_i)\}_{i=1}^n$ drawn iid from the training distribution P_{tr}. Then for every $f \in \hat{F}$ and $\delta > 0$, with probability at least $1-\delta$*

$$R[P_{te}, \hat{f}] - R[P_n^w, f] \leq \frac{1}{\sqrt{n}} \left(2LR + LRW\sqrt{2\ln(2/\delta)} + |L(0)|W\sqrt{2\ln(2/\delta)}\right) \tag{2.9}$$

Proof. For all $f \in \hat{F}$

$$R[P_{te}, f] - R[P_n^w, f] \leq \sup_{h \in \tilde{L} \circ F} (R[P_{te}, f] - R[P_n^w, f])$$
$$L(0)\left(1 - \frac{1}{n}\sum_{i=1}^n w(x_i)\right),$$

using the fact that $L(y, y') = L(yy')$ is L-Lipschitz and applying Mcdiarmid's inequality to the random variables $\sup_{h \in \tilde{L} \circ F} (R[P_{te}, f] - R[P_n^w, f])$ and $\left(1 - 1/n \sum_{i=1}^n w(x_i)\right)$ we will get with probability at least $1-\delta$

$$\begin{aligned} R[P_{te}, f] - R[P_n^w, f] \leq E_{tr} \sup_{h \in \tilde{L} \circ F} (R[P_{te}, f] - R[P_n^w, f]) + \\ LRW\sqrt{2\ln(2/\delta)/n} + |L(0)|W\sqrt{2\ln(2/\delta)/n} \end{aligned} \tag{2.10}$$

The same argument as in the proof of theorem 8 [22] with the teorem 12 part 4 [22] and Appendix B [19] gives us

$$E_{tr} \sup_{h \in \tilde{L} \circ F} (R[P_{te}, f] - R[P_n^w, f]) \leq \mathcal{R}\left(\tilde{L} \circ F\right) \leq L\mathcal{R}(F) \leq \frac{1}{\sqrt{n}} LR. \tag{2.11}$$

Considering (2.11) in (2.10) finishes the proof. □

3. Network Simulation with Generative Adversarial Networks

Evaluation of network policies in 5G architectures is becoming a daunting task as the networks' complexity and heterogeneity grows. experimental measurements require expensive setup and dedicated campaigns [23, 24], and analytical models are left behind due to multiple simplifying assumptions that limit their applicability. Although Computer aided simulations are commonplace in the research community, they also suffer certain limitations. For instance

- 5G networks require high resolution synthesis of channel propagation that incorporates the spatial consistency and temporal evolution of the network. While ray tracing is a common method to model the channel accurately [24], its complexity grows exponentially with the transmitters/receivers and the obstacles in the environment, prompting the use of additional systems/data for motion detection and positioning systems (e.g. LIDAR [25, 26], SUMO [27]) and limitimg its application to simplistic scenarios [28] to reduce the computation overhead.

- End-to-end optimization of the cellular networks requires end-to-end evaluation of the systems [29–34] and accurate simulation of the full network stack and many heterogeneous components in the path, including the generated traffic from user equipment/devices, new radio (NR) [25, 26], 5g core, multi-radio access technology (RAT) protocol [35], traffic management and the overhead of SDN/NFV systems [36]. In current simulators, individual components and their interactions with each other are hard coded, therefore keeping them up-to-date in ever-changing 5G networks is a time consuming process and demands specialization.

- While ML and deep learning is a promising venue to enhance the state-of-the-art performance of cellular networks, one of the main challenges for their adaptation is their demand for large amount of data. Training datasets should cover multiple diverse scenarios and be parametric for the required pre/post processing of data. Existing prototype measurements [37] and dataset generation methods lack the scalability and flexibility and heavily rely on the abstract models that are currently used for simulation [27, 38],

- Despite the vast difference between 4G and 5G networks and the considerable number of introduced novel technologies with various demands, our approach to simulation remains mainly unchanged and heavily depends on sampling from abstract models [23, 39–41].

Generative deep models, particularly Generative Adversarial Networks (GANs), are able to generate high dimensional complex distributions and can help to overcome the mentioned challenges. Although it is theoretically proven that GANs are universal simulators, their performance and limitations in practice with limited resources and training data is not fully understood. Hence, in order to use the adversarial framework in communication systems, we first need to investigate the following:

- How can we evaluate models that are trained using adversarial framework? It is important to know which neural network architecture or training algorithm generates more realistic data for certain tasks. However, evaluation of GANs remains an open problem to this day. Many works propose evaluation of different proxy measures such as Inception score [42], modified Inception score [43], mode score [44], Frechet Inception distance [45] or other methods such as precision and recall [46], approximation of likelihood [47, 48], or its lower bound [49, 50] to tackle this problem however, none completely solves this. [51, 52]

- How to compare different simulators? Instead of a single model evaluation methodology, comparative evaluation aims to identify the generative distribution that is a better match for a real distribution given a set of generative models. This perspective relaxes the model evaluation problem and allows us to use two sample tests from statistical analysis [53].

- How does the performance of some arbitrary application on simulated data reflect the performance on real data? While synthesized samples where previously used for data augmentation and robustness, to our knowledge, there is no study on the performance of models that are solely trained on synthesized data. Recent works suggest the existence of memorization phenomena in over-parameterized neural networks, e.g. CNNs [54, 55] and auto-encoders [56]. It is important to quantify how memorization affects GANs in order assess its impact on the successive applications.

The rest of this chapter is organized as follows: in section 3.1, we concisely introduce generative models that are commonly used in practice, showing how GANs compare to these models and motivating the use of adversarial framework for network simulation. We give a detailed description of GAN architecture and adversarial training and it's various extensions in section 3.2. In section 3.3, we discuss the existing GAN evaluation methods and point out their strengths and deficiencies. Finally, we explain the memorization phenomena in section 3.4 and explain possible ways to exploit it for adversarial training.

3.1. Generative Models

Generative models are powerful tools for learning data distributions and synthesizing samples. In statistical machine learning, these models are trained on limited number of training data samples to learn the joint distribution P_{model} that models the distribution of observed random variables $\mathbf{x} = (x_1, x_2, ..., x_d) \in \mathbb{R}^d$. We denote the true distribution with P_{data} and its probability density p_{data}. The goal is to draw samples from P_{model} or evaluate its density explicitly on the underlying data manifold. There are three types of generative models listed below.

3.1.1. Fully Observable Models

They model the dependencies of continuous or discrete random variables $\mathbf{x} = (x_1, x_2, ..., x_d) \in \mathbb{R}^d$ where all variables are assumed to be observed. In the most general case we can evaluate the joint distribution density $p_{\text{data}}(\mathbf{x})$ by modeling the factors in the chain rule below, e.g.

with parametric functions.

$$p_{\text{data}}(\mathbf{x}) = p_{\text{data}}(x_1) \prod_{i=2}^{d} p_{\text{data}}(x_i | x_{1:i-1}) \tag{3.1}$$

In Bayesian networks [57], ordering the random variables based on domain knowledge and considering their independencies reduces the computation cost exponentially. In the induced directed graphical model, we can optimize the parameters of the model density $p(\mathbf{x})$ directly by maximizing the likelihood of of hold out data. However, they are sensitive to ordering and sample generation is sequential and very slow process in high dimensions.
In undirected graphical models (e.g. [58, 59]), $p(\mathbf{x})$ is proportional to the product of energy functions. For instance, if x_1 and x_2 depend on each other, then the factorization of $p(\mathbf{x})$ includes the energy function $f(x_1, x_2)$ which can be parameterized by a DNN. While undirected graphical models are flexible and don't induce an ordering on variables, evaluation of likelihood $p(\mathbf{x})$ is often intractable since we don't have access to its normalization factor. Therefore, training these models often requires methods such as contrastive divergence [60] or score matching [61] that avoid likelihood estimation. Also, sampling from these models is not straight forward and require Markov Chain Monte Carlo (MCMC).

3.1.2. Transformation Models

In transformation models, a latent random variable $\mathbf{z} \in \mathbb{R}^k$ is drawn from $p(\mathbf{z})$ with know density and *deterministically* transformed to a continuous observable $\mathbf{x}$ using a parameterized function $f : \mathbb{R}^k \to \mathbb{R}^d$. If the transformation is an invertible function, the model is called a Normalizing Flow and the likelihood $p(\mathbf{x}) = p(\mathbf{z}) |\det \frac{\partial f}{\partial \mathbf{z}}|^{-1}$ can be evaluated easily if function f is simple and the det can be computed efficiently. Therefore, the choice of transformation is rather restricted. In addition, we have the constraint on the latent space dimension $k = d$ to ensure invertibility of f or $k \geq d$ for variational evaluation of the likelihood [62].

3.1.3. Latent Variable Models

In latent variable models, a latent variable $\mathbf{z} \in \mathbb{R}^k$ is introduced that represents the high level causes of the observation $\mathbf{x}$. In this case conditional $p(\mathbf{x}|\mathbf{z})$ is modeled by a simple parametric distribution. Even using a simple prior $p(\mathbf{z})$ and conditional, these models can realize complex dependencies between observed variables and they easily generate new samples [63–66]. Furthermore, the variable $\mathbf{z}$ provides a low dimensional representation of the data and can be used to generate samples with desired properties. However, evaluation of marginal likelihood $p(\mathbf{x})$ for model scoring, ccomparison and selection is intractable since we can't evaluate the following integral unless $p(\mathbf{x}|\mathbf{z})$ has a simple form and we can analytically evaluate the following integral.

$$p(\mathbf{x}) = \int p(\mathbf{z}) p(\mathbf{x}|\mathbf{z}) d\mathbf{z} \tag{3.2}$$

As a result, training these models requires a variational approximation of inference probability, $p(\mathbf{z}|\mathbf{x})$, maximization of a lower bound of likelihood as was done in [63–65]. While Variational models such as Variational Autoencoders (VAE) are easier to train and provide a lower bound on the marginal likelihood, their samples look unrealistic and fuzzy.

Another alternative is adversarial learning used for training GANs, where the generated samples are discriminated from the data samples using a statistical two sample test performed by an adversary. This approach avoids evaluation of the intractable marginal likelihood $p(\mathbf{x})$ and its approximate variational lower bounds and it is shown that the generator recovers the data distribution under non-parameterized setting. In practice, with parameterized generator and discriminator, GANs generate high resolution, complex and realistic samples, while their success in other domains remains limited as there is no standard metric for their evaluation. A more detailed description of GAN training is provided in section 3.2.

3.2. Generative Adversarial Networks

Adversarial framework consists of two players: a generator $G : \mathbb{R}^k \to \mathbb{R}^d$ that is a *deterministic* mapping from latent space to data space, and a discriminator $D : \mathbb{R}^d \to \mathbb{R}$ that separates real and fake data. Samples are generated by drawing samples from prior $\mathbf{z} \sim p_z$ which typically has a standard gaussian distribution and transforming them through the generative map G. This model can be categorized as a transformation generative model or a latent variable model with a gaussian $p(\mathbf{x}|\mathbf{z}) = \mathcal{N}(G(\mathbf{z}), \sigma\mathbf{I})$ where $\sigma \to 0$. Therefore, unlike Normalizing Flows and variational models, it can realize degenerate distributions where data lies on a low dimensional manifold of the d-dimensional space and it is capable of generating realistic samples with fine details.
The training is a sequential minimax game where generator tries to fool the discriminator by generating realistic samples and minimizing the value function on equation 3.3, while the discriminator maximizes the binary cross entropy objective to distinguish fake samples $\mathbf{x} \sim p_g$ generated by the generator and data samples drawn from the true distribution, $\mathbf{x} \sim p_{\text{data}}$.

$$\begin{aligned} V(D, G) &:= \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}} [\log D(\mathbf{x})] - \mathbb{E}_{\mathbf{z}\sim p_z} [\log 1 - D(G(\mathbf{z}))] \\ &= \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}} [\log D(\mathbf{x})] - \mathbb{E}_{\mathbf{x}\sim p_{g_\theta}} [\log D(\mathbf{x})] \end{aligned} \tag{3.3}$$

The minimax objective is given below.

$$G^*, D^* = \underset{G}{\operatorname{argmin}} \underset{D}{\operatorname{argmax}}\, V(D, G) \tag{3.4}$$

In [66] it is shown that in the non-parameterized setting, the optimal discriminator for a given generator $D^*(G)$ is the bayesian classifier $\frac{p_{\text{data}}}{(p_{\text{data}}+p_g)}$ and the optimal generator is minimizing the Jensen-Shannon divergence between p_{data} and p_g and therefore at the critical point we have $p_g = p_{\text{data}}$. If we parameterize the generator and discriminator with DNNs and obtain G_θ and D_ϕ, respectively, an alternating Stochastic Gradient Descent Ascent algorithm can be used to converge to the saddle point of $V(D_\phi, G_\theta)$ avoiding the computation of the intractable likelihood in equation 3.2 for training.
f-GANs generalize the objective in equation 3.3 with generic f-divergences. In [67] the discriminator is replaced with a critic and variational lower bound of f-divergence of p_{data} and p_g is obtained using Fenchel conjugate of f. However, as it was observed in [68], when the data distribution lies on low dimension manifold, it is unlikely to have an intersection of the data and model manifolds with non-zero measure and f-divergences are non-continuous in the parameter space. Consequently training GANs with f-divergences is unstable and can result in oscillations of the objective. To solve this [68] proposes WGAN, modifying the GAN

objective with a critic that maximizes the dual of Wasserstein distance of the distributions and [69] recommends to penalize the gradients of critic to relax the optimization constraint in the dual Wasserstein objective. WGANs are more stable and the generator's objective doesn't saturate even if the critic has a more powerful architecture. In addition, Wasserstein distance can be used to track convergence during training since it correlates with visual fidelity of image distributions.

Training GANs with iterative optimization algorithms has illustrated multiple problems. One important failure of GANs is the mode dropping or mode collapse problem, where the generator learns a few major modes of the data distribution and doesn't generalize to the rest of the manifold or multiple input values $\mathbf{z}$ are mapped to the same output and the model ends up assigning small probability measures to majority of the support of the data. To enforce diversity of the generated samples, we can use an auxiliary inference network $E : \mathbb{R}^d \to \mathbb{R}^k$ to reconstruct the samples and add the mode regularizer $\lambda \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}} [\log D(G(E(\mathbf{x})))]$ to the generator's loss. Since the reconstructed samples are more accurate in the major modes, the regularizer reduces the contribution of major modes to the gradient signal and the generator generalizes to the less visited regions of the data manifold. Another approach is to regularize the objective with the entropy of the model distribution, $\lambda H(p(\mathbf{x}))$, to encourage models with larger entropy. The entropy is not well-defined in general, therefore in [70], the authors propose to add gaussian noise to generator's output and obtain an unbiased estimate of the entropy using MCMC.

Another problem is overfitting to the training dataset and memorization. While there is no conclusive measure to demonstrate memorization, recent works suggest that when DNNs are over-parameterized, they use their excessive capacity to store information that hurts performance for worst group while improving the average performance [71]. A similar problem is observed in over-parameterized autoencoders in [56] which we will elaborate on in section 3.4.

3.3. GAN Evaluation

In this section we summarize some of the main methods adopted by the research community to evaluate and benchmark GANs, while pointing out the strengths and drawbacks of each approach. We only mention methods that can be adapted to simulation of communication systems. We refer the interested readers to [51] for a more comprehensive review of GAN evaluation methods.

3.3.1. Visual Fidelity

Inception Score (IS) was proposed by Salimans et al. [42] and it evaluates the visual fidelity of generated samples from labeled datasets. We denote the label variable with y. IS measures the average KL-divergence between the conditional label distribution $p(y|\mathbf{x})$ of generated samples and marginal distribution $p(y)$ where the labels are assigned using an Inception Net classifier [72] that was pre-trained on ImageNet [73].

$$\text{IS}(p_g) = \exp(\mathbb{E}_{\mathbf{x}\sim p_g} [D_{\text{KL}}(p(y|\mathbf{x})||p(y))]) \tag{3.5}$$

This measure shows a reasonable correlation with sample quality and diversity, however, it is insensitive to overfitting. In addition, it is believed that IS cannot reliably detect if the model

is dropping minor modes of the distribution [51]. To favor models with higher diversity per each category, Gurumurthy et al. propose Modified Inception Score (m-IS) that measures average KL-divergence between conditionals $p(y|\mathbf{x}^{(i)})$ and $p(y|\mathbf{x}^{(j)})$ where $\mathbf{x}^{(i)}$ and $\mathbf{x}^{(j)}$ are samples from the same class [43].
IS and m-IS don't account for similarity of the model to data distribution. This is addressed in Mode Score where the marginal label distribution of both distributions are compared.

$$\text{m-IS}(p_g, p_{\text{data}}) = \exp(\mathbb{E}_{\mathbf{x}\sim p_g}[D_{\text{KL}}(p(y|\mathbf{x})||p_{\text{data}}(y))] - D_{\text{KL}}(p(y)||p_{\text{data}}(y))) \tag{3.6}$$

where $p_{\text{data}}(y)$ is the empirical distribution of labels of training dataset [44]. AM Score takes the imbalanced classes of training dataset into account as it requires the classifier to be trained on the given dataset.

$$\text{AMScore}(p_g, p_{\text{data}}) = -\mathbb{E}_{\mathbf{x}\sim p_g}[H(p(y|\mathbf{x}))] - D_{\text{KL}}(p(y)||p_{\text{data}}(y)) \tag{3.7}$$

Frechet Inception Distance (FID) is introduced by Heusel et al. where embeddings of real and fake data samples are obtained from layers in Inception Net and two gaussian distributions with mean μ_{data} and μ_g and covariance Σ_{data} and Σ_g are fitted to the embeddings. The Frechet distance between these two gaussian distributions quantifies the quality of generated samples.

$$\text{FID}(p_g, p_{\text{data}}) = \|\mu_{\text{data}} - \mu_g\|_2^2 + \text{Tr}(\Sigma_{\text{data}} + \Sigma_g - 2(\Sigma_{\text{data}}\Sigma_g)^{1/2}) \tag{3.8}$$

FID Score is partially sensitive to mode invention and mode dropping and noise in the samples as it measures a distance between the embeddings of generated and data distributions. While visual fidelity methods can be adopted for communication system simulation, their main drawback is that they are only applicable for labeled datasets and their performance depends on the classifier and the classification task. Obtaining labels for unsimulated datasets obtained from measurements in wireless network is expensive [23, 24].
Some of the other approaches that rely on a classifier that is trained on real or fake samples are proposed in [74–80]

3.3.2. Log-likelihood Estimation

Kernel Density Estimation (KDE) or Parzen window is a well-established method for estimating the density function of a distribution given its samples. Using a normalized probability kernel K and i.i.d. samples $\{\mathbf{x}^{(1)}, ..., \mathbf{x}^{(N)}\}$, the density is approximated by the uniform mixture $\hat{p}(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} K(\mathbf{x} - \mathbf{x}^{(i)})$. This approach leads to biased estimations in high dimension and requires very large number of samples to achieve acceptable accuracy. Since this approach produces rankings different from the other measures, it has been questioned by Theis et al. [52]
Wu et al. [47] proposed to use Annealed Importance Sampling (AIS) [81] to approximate the log likelihood given in equation 3.2. They assume a gaussian observation model $p(\mathbf{x}|\mathbf{z})$ with mean given by the map G and fixed variance. They evaluate log marginal likelihood with two methods which for simulated models correspond to stochastic lower bound and upper bound of the log likelihood. The key drawback of this method is the fixed variance of gaussian observations which is biased and favors models with large variances.
In [70], Dieng et al. also propose to add gaussian noise to the observations so that the derived generative model is similar to VAEs with gaussian $p(\mathbf{x}|\mathbf{z})$ with parameterized mean

and covariance matrix. Addition of the noise results in well-defined yet intractable entropy $H(p(\mathbf{x}))$ and log marginal likelihood. This approach can recover all the modes of the data distribution in their experiments. To estimate the marginal likelihood, the authors propose training an encoder at test time and using importance sampling with samples drawn from the encoder, which is prone to high variance.
Marginal likelihood is frequently used as a measure for training and evaluating deep generative models. Optimization of log likelihood in generative and discriminative models is equivalent to minimization of the KL-divergence of data and model distributions. At the critical point this optimization results in complete matching of generative and data distributions. However, evaluating the log likelihood alone can be uninformative since we don't know how far away the model is from the true distribution. As it was shown by Theis et al. sample quality doesn't correlate with model likelihood and a model with arbitrarily low quality samples can have high likelihood [52]. Sanchez-Martin et al. in [48] show that this results in skewed likelihood estimations where samples that are closer to the model are assigned higher likelihood regardless of their true probability density. Therefore, likelihood estimation methods for model scoring, selection and comparison should be adopted with cautious. In order to solved this, the authors of [48] propose to evaluate the likelihood of reconstructed samples instead of test data, therefore separating the reconstruction error caused by the model mismatch using the following equation.

$$p(G(E(\mathbf{x})) = \int p(G(E(\mathbf{x}))|\mathbf{z})p(\mathbf{z})d\mathbf{z} \tag{3.9}$$

where E is an encoder network trained to reconstruct the samples as in the BiGAN [82] and MDGAN [44] architectures. The reconstruction likelihood is estimated with importance weighting.

3.3.3. Hypothesis Testing and Probability Distance Measures

Two sample tests are designed to identify the correctness of null hypothesis which is $P_{\text{data}} = P_{\text{model}}$ from their corresponding datasets [53].
Maximum Mean Discrepancy (MMD) measures dissimilarity of two distributions from their samples [83]. The kernel MMD uses a symmetric characteristic kernel function K such as Gaussian kernel and measures MMD of p_{data} and p_g with

$$\text{KernelMMD}(p_g, p_{\text{data}}; K) = \mathbb{E}_{\mathbf{x},\mathbf{x}'\sim p_{\text{data}}}\left[K(\mathbf{x},\mathbf{x}')\right] + \mathbb{E}_{\mathbf{x},\mathbf{x}'\sim p_g}\left[K(\mathbf{x},\mathbf{x}')\right] - 2\mathbb{E}_{\mathbf{x}\sim p_{\text{data}},\mathbf{x}'\sim p_g}\left[K(\mathbf{x},\mathbf{x}')\right] \tag{3.10}$$

The Wasserstein critic in WGAN provides the Wasserstein distance as a measure of the distance between real and generated distributions [68]. The distance is given below for a critic f that is 1-Lipschitz.

$$\text{W}(p_g, p_{\text{data}}) = \max_f \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}}[f(\mathbf{x})] - \mathbb{E}_{\mathbf{x}\sim p_g}[f(\mathbf{x})] \tag{3.11}$$

Probability distance metrics and two sample tests are able to reliably detect mode collapse and overfitting as they compare the density functions of the generative and data distributions. However, they are computationally expensive and have a high sample complexity.

3.3.4. Coverage and Outlier Measurement

Birthday test approximates the size of the support of generated distribution by identifying duplicates in a set of generated examples. This test measures diversity and mode collapse but is independent of the true distribution [84].
Coverage metric measures the probability mass of the real data that is covered by the model distribution is defined as $C := P_{\text{data}}(dP_{\text{model}} > t)$ where t is the largest value that $P_{\text{model}}(dP_{\text{model}} > t) = 0.95$. KDE is used to approximate the density of P_{model} [85].
Sajjadi et al. propose to use precision-recall curves to quantify the quality of generated samples as well as the proportion of data distribution that is covered by generated distribution indicating coverage of the modes [46]. The authors show that there is a trade-off between precision of the generated samples and recall of the true distribution and a certain level of precision corresponds to a certain recall value. These measures generalize the total variance on state space of the union of the distributions Ω. For a given $\lambda \in (0, \infty)$, precision and recall of a distribution P_{model} wrt to P_{data} is given below.

$$\text{Precision}(p_g, p_{\text{data}}; \lambda) = \sum_{\omega \in \Omega} \min(\lambda P_{\text{data}}(\omega), P_{\text{model}}(\omega)) \tag{3.12}$$

$$\text{Recall}(p_g, p_{\text{data}}; \lambda) = \sum_{\omega \in \Omega} \min(P_{\text{data}}(\omega), \frac{P_{\text{model}}(\omega)}{\lambda}) \tag{3.13}$$

The set of precision and recall values for all $\lambda \in (0, \infty)$ can be approximated by clustering the union of real and fake samples in feature space and computing precision and recall over discretized clusters. This approach is also a generalization of the method proposed in [86] by Richardson et al. where a statistical test on similarity of the distributions is derived by computing the average difference of the number of real and fake samples that fall into each cluster.
Huang et al. in [87] propose comparing the rate-distortion curves to assess mode dropping and invention of decoder based generative models. A variational upper bound of the curve can be approximated with AIS illustrating the asymptotic minimum distortion level which measures model mismatch and the achievable rate of lossy compression of the data using the model distribution, which for a given model $p(\mathbf{x}|\mathbf{z})$ is controlled by the prior. As opposed to precision-recall measure, rate-distortion curves can evaluate the difference of the models even in the parts of the manifold that the support of generative and data distributions don't intersect.

3.4. Memorization in Over-parameterized Models

Over-parameterized DNNs have disproved the previous beliefs in that models that can interpolate the training data don't generalize. Belkin et al. in [88] show that contrary to the traditional statistical learning models, hypothesis classes with larger capacity can achieve smaller generalization gap (i.e. the distance between average train and test losses) compared to simpler hypothesis classes. This supports the conjecture that training over-parameterized DNNs with algorithms like Stochastic Gradient Descent (SGD) induces an implicit bias on the types of the functions that DNNs represent, although in principle these can represent functions with more complex structures.

Understanding the inductive bias in over-parameterized DNNs is currently a hot topic of research. Discovering the main cause of this bias is important since it can uncover the existing problems with entangled representations and non-causal learning. As was illustrated by [71] overparameterized DNNs tend to memorize the features of the minor modes of the distribution and therefore minimize the average loss while not actually learning the discriminative distribution of the worst group of samples. Hence, it is important to understand how to control or even exploit the memorization to our advantage.
Quantifying the memorization for learning algorithms is not a trivial task. However, for fully connected and convolutional autoencoders, Radhakrishnan et al. in [56] show that SGD is biased toward hypotheses that are locally contractive at the training examples. To show this the authors iteratively apply the autoencoding on noise samples and retrieve the training samples after a few iterations. As it turns out, depth is an important factor in the strength of the contraction. Multiple generative models have a similar encoder-decoder architecture, however with stochastic models instead of deterministic mapping as in autoencoders. For instance for variational autoencoders [63] and bidirectional GANs [44, 48, 82] an approach that is similar in spirit to iterative retrieval in [56] may clarify the true hypothesis class of SGD and guide us to design algorithms to avoid overfitting and memorization in GANs.

3.5. Conclusion

Deep generative models have shown tremendous success in domains such as computer vision and natural language processing in various tasks such as semi-supervised learning, representation learning, missing data imputation, in-painting, denoising, generating realistic and complex samples from natural data distribution and density approximation.
Especially the recent success in synthesis of high dimension complex distributions with GANs promises their successfull adaptation in other domains such as wireless network simulation. While the prospect of simulating wireless problems with GANs is exciting, we should make sure that we can evaluate their performance and carry out certain tasks such as model scoring, comparison and selection through reliable measures, in order to understand and manage the likelihood, diversity and accuracy of the simulations.

4. Machine Learning applied to MIMO

Since the third-generation of wireless mobile telecommunication (3G) multiple-input and multiple-output (MIMO) systems have been used to enhance communication performance. With the increasing numbers of receivers, massive MIMO (using up to hundreds of antennas) becomes a natural extension of the MIMO technology. Studies [89] indicate that, when proper interference-suppressing precoding is applied, the maximum energy efficiency is achieved when the number of antennas is around 100–200. In a scenario with hundreds of antennas, many of the currently available solutions become unfeasible as their execution time becomes too large to be deployed in real-world applications.

Currently deployed small-scale MIMO systems typically contain 2–4 antenna receivers [90]. This is in stark contrast with massive MIMO (mMIMO) architectures, where antennas arrays are considered to be in the order of hundreds. Resource allocation becomes a challenging matter in such large scale scenarios. Furthermore, multi-antenna receiver implementations often require expensive operations, such as matrix inversion. When considering large amount of antennas and users scalability and feasibility become an issue. Similarly, solutions designed to work iteratively become hard to handle as their execution time often do not scale linearly.

This Chapter focuses on the applications of machine learning solutions to massive MIMO (mMIMO) and the challenges originated from this new paradigm. Generally, machine learning solutions are applied to address model- or algorithm-deficient problems, as illustrated in Figure 4.1. General physical-based mathematical models (Figure 4.1a) require extensive research over a specific domain which then can lead to a designed solution [91]. Typically, when this solution is not feasible (Figure 4.1b), machine learning solutions benefit from learning hypothesis classes originated from the mathematical model [92–94]. In the worse case, when there is a lack of domain knowledge (Figure 4.1c), models are trained based on a generic optimization scheme [95–97].

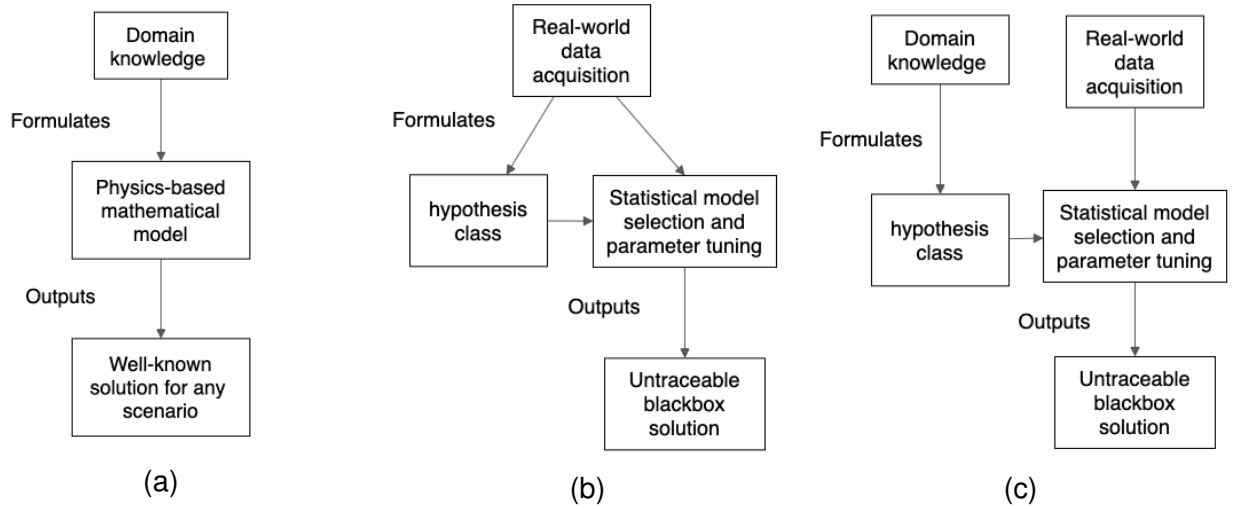


Figure 4.1: Comparison among different ways to develop a model. (a) Conventional engineering design flow. (b) Baseline machine learning methodology. (c) Integration of domain knowledge during model training.

We proceed by discussing different challenges when applying machine learning in model- or algorithm-deficient problems. We start by discussing recent trends on data-driven solutions proposed for mMIMO in wireless communications and the new challenges imposed by those solutions. In the following sections, we focus on resource allocation in cell-free MIMO, beamforming selection and clustering users for MAC and broadcasting channels.

4.1. Scalable Cell-Free mMIMO

As opposed to conventional collocated massive MIMO, where antennas are located in a small and compact area, cell-free mMIMO follows a distributed architecture. Generally, cell-free mMIMO systems assume a large number of service antennas to be in a widespread area serving a much smaller number of users [98], i.e., a high antenna–user ratio is assumed. This distributed setting allows a greater coverage and higher throughput rates when compared to the conventional approach [99]. At the same time, this new perspective poses a number of novel challenges, such as enhanced pilot contamination [97], complex resource allocation [100], increasing demand over backhaul [89, 99], and process coordination [99, 101]. Following the new trends in the research community, many academic works advocate for a data-driven approach to address such challenges.
Conventional systems handle pilot contamination by assigning quasi-orthogonal pilot sequences to different terminals, thus, helping mitigate inter- and intra-cell interference. However, as the number of users and access points (AP) increases, the task of assigning quasi-orthogonal pilot for each user becomes more challenging. In cell-free mMIMO systems this problem is even more serious since, even under the assumption of limited user–AP associations, such assignments become barely tractable due to the distributed nature of the AP as opposed to conventional co-located systems. A straightforward solution would be to deploy random pilot assignment [102]. More robust approaches have considered structured pilot assignment [103] and assignment based on prior clustering users according to their channel properties [104].
Closely related, power allocation strategies play a critical role in the task of reducing interference among different users and access points. Cell-free massive MIMO requires strategies that can be implemented in a distributed manner and with scalability properties in order to manage a large number of users and access points. As each AP only partially observes the network, it becomes hard to compute a globally optimal solution in a distributed fashion. At the same time, a centralized processing solution becomes unfeasible [102]. State-of-the-art approaches either try to solve a global max–min optimization problem [99, 105] or design a scalable algorithm at the cost of performance loss [100].

4.1.1. Data-driven Approaches and Their Imposed Challenges

As argued before, both power control and pilot assignment problems often pose themselves as either model or algorithmic deficit. Solutions available for current technologies often become unfeasible due to high number of parameters, model constraints or long optimization procedures. Yet, such systems require efficient optimization techniques to handle the different sources of interference and provide high-quality service. Such characteristics intuitively lead to data-driven attempts to deal with such ill-posed problems. Addressing this idea, prominent works have proposed the use of low-complexity artificial neural networks to mimic

complex algebraic solutions [91]. Training is performed offline, which reduces the problem complexity at execution time but raises questions regarding its efficiency in real-world applications. As a matter of fact, it has been shown that models trained and tested in different distributions tend to dramatically reduce their correctness [106]. Another drawback from this approach relates to scalability issues. By using a fully connected neural network, it becomes necessary to train different models for different wireless settings and store them for further reuse.

Fully convolutional neural networks do not suffer from the same drawback as each neuron receptive field is only a portion of the previous layer rather than entire layer as in fully connected networks. This characteristic of fully convolutional networks is taken into account in [107], where the authors suggest a two-layers convolutional neural network to perform joint and data power control. As a result, the solution requires training once and allows prediction in different settings, with different number of users.

These works show how machine learning approaches can be used in order to promote further advances in cell-free massive MIMO. At the same time, applying data-driven solutions comes with inherited challenges. The easiest one to spot is the lack of reliable data. At the physical layer (PHY), it becomes expensive to acquire real-world data, which is usually experimentally dealt with using synthetic data drawn from different channel models and simulators [108]. Recent works have suggested that deploying models trained only in synthetic data is non-optimal and lead to a substantial performance loss in wireless communication applications [106]. At the same time, promising researches advances, in the field of natural language processing (NLP), suggest that embedding mechanisms are of major importance to aid on language translation [109] and scene description [110, 111]. Moreover, leading research works in one–shot learning apply transfer learning techniques to reduce training time [112]. Perhaps such researches could guide further research in highly adaptive models, capable of quickly adjusting to the different over-the-air communication characteristics.

All things considered, machine learning solutions become hard to track and lead to unknown performance predictions. In other words, without the knowledge of the training set, it becomes hard to understand the lower and upper bound of the solution and guarantee certain service requirements in terms of performance. Furthermore, in wireless communications, service requirements depend on the application and pre-defined specifications. Accounting for those different scenarios might be a hard task for data-driven models. To train a reliable model, it has to be trained and validated with sufficient and meaningful data acquired from different scenarios, e.g., number of users, fading and power assignments.

To approach such challenges, trying to understand what guided a neural network to choose a specific solution has become an active line of research in the machine learning community [113,114]. In computer vision, this study is normally done using attention mechanisms [115] which contribute to visual results assimilation. The same principle can be extended to text analysis [111]. The wireless network community has also contributed to further interpret how deep learning-based models understand wireless channels [116]. In this work, the authors argue that using ReLU activation functions together with fully-connected layers leads to approximations on the minimum mean-square error (MMSE) solution. That is because a fully connected neural network with ReLU activation mirrors piecewise linear functions. This promising result inspire future works by considering different types of architectures and activation function.

4.2. Beam Selection

High-frequency bands have raised significant interest in the research community. Due to the potential for high rate bandwidth, mmWaves in the spectrum between 30 and 100 GHz are targeted as promising frequencies in 5G technologies [117]. On the one hand, the strong propagation losses of these bands motivate the use of large antenna arrays steering very narrow beams, which makes the transmission highly directional and sensitive to blockage. Channel characteristics are also very different from commonly used microwave frequency. On the other hand, thanks to the high frequencies, mmWave arrays can be built in very small areas, allowing for a higher number of antenna elements and increasing the overall spatial gains. For all these reasons, systems operating in such narrow frequencies are highly dependent on beamforming technologies at both transmitter and receiver side. Thus, leading to beam selection in order to create meaningful and reliable communication gains.

In both mmWave and sub-6GHz transmissions, the design and implementation of massive MIMO digital receivers can be a difficult task due to the number of inherent operations involved. The complexity of commonly used algorithms, such as weighted minimum mean square error (WMMSE), grows due to large matrix inversions and discrete variables. In this context, machine learning solutions have raised special attention among different research groups. Using unsupervised learning, Huang et al. [118] proposes a fast beamforming for downlink MIMO. The learning police considers maximizing the weighted sum-rate of all users according to learned transmit filters. Finally, the authors suggest using pruning techniques to reduce the network size and enhance complexity performance during validation.

Despite having different propagating characteristics, exploiting sub-6 GHz channel information might bring remarkable contributions to mmWaves. As 5G and beyond 5G networks are expected to operate in both bands, exploiting the former's channel information in support of the latter has become an active line of research. With that in mind Sim et. al. [117] employ channel characteristics, specifically the power delay profile, of sub-6 GHz band in a deep learning model to perform beamforming selection in the mmWave band. Typically, deep learning-based models are applied in such a situation to reduce the search space. Results are validated in both 3D-ray-tracing simulations and over-the-air experiments.

Further research in the autonomous driving industry suggests that mmWave beam selection plays a key role in sharing sensor data for connected vehicles [119]. In this context, machine learning has proved to be useful in the detection of line-of-sight (LoS) or non-line-of-sight (NLoS) signals and in the beam selection problem [108]. The approach considers a downlink orthogonal frequency-division multiplexing (OFDM) mmWave system and trains a deep neural network to output top-K ranking for different beam patterns. Data is generated using a ray-trace model which considers different mobility traffic and static scenarios.

Machine learning might be the key to many challenges for communication in the mmWave spectrum. As strong propagation losses are common in this spectrum, data-driven solutions might bring new insights in how to mitigate this effect. Moreover, it is notable that machine learning solutions also bring new challenges for the task at hand. A key point on developing robust solutions lies on acquiring reliable and meaningful data, capable of represent the highly dynamic environments in which are expected to be implemented.

4.3. Clustering Users for MAC and Broadcasting Channels

Radio access technologies are widely studied as means to enhance wireless communication spectral efficiency and connectivity. In the uplink, non-orthogonal multiple access (NOMA) has recently become a key mechanism to significantly enhance communication rates. By exploring users channel quality and sharing time/frequency slots with a group of users, it becomes possible to detect multiple simultaneous transmissions via successive interference cancellation (SIC). Other approaches rely on jointly interference cancellation methods [120]. Either way, an important problem is to decide which users should share the available resources in order to maximize the total sum rate.

Downlink transmissions are much more challenging due to the unavailability of channel state information at the transmitter, which is the entity that performs spatial processing. In practice, one can hardly rely on reciprocity, either because uplink and downlink take place at different frequencies (FDD) or because they use different frontends (TDD). The insertion of pilots and the use of a feedback channel can help in overcoming this lack of CSI, but at the cost of substantial rate losses.

In the uplink, combining user clustering and NOMA strategies plays a key role for efficient usage of the link. Figure 4.2 illustrates this idea. A base station (BS) shares each resource among a group of users with different channel qualities (Figure 4.2a). In the example, this quality difference is generated by different distances from the BS. Users grouped together can superimpose each other's signals; if the signal strength is sufficiently different among users in a group, they can be retrieved at the receiver side via SIC processing. This mechanism allows users that are spatially close to one another to share the same resource at the same time and helps increase energy efficiency [121]. For users that are well spaced in the angular space domain, it becomes easier to separate them via beamforming (Figure 4.2b). The question remains on how to define well-spaced users in the angular space domain. As it turns out, clustering users according to their channel characteristics and spatial position is an active line of research.

In [121] user pairing is considered for the case multi-antenna but proposes jointly decoding different groups together using Hungarian algorithm. As the number of transmitters and receivers increases, this decoding strategy becomes more costly than conventional successive interference cancellation. The work in [92] addresses a similar matter using a neural network trained to predict user-cell association. In particular, every transmitter location is assumed to be known by all the receivers. A relaxation of the original problem is solved by standard linear programming and is used as training data for the neural network. However, the solution is not scalable due to the neural network architecture. Nonetheless, the work exemplifies how machine learning can bring a new perspective to the problem.

In many situations, the exact CSI is difficult to acquire but long term statistics are easy to estimate. Along these lines, Joint Spatial Division and Multiplexing (JSDM) schemes propose dividing users into groups based on second-order channels statistics and their eigenvectors [122]. Users are served according to a two-stage precoding scheme, where one of the stages is built using only second order statistics whereas the other uses the estimated instantaneous CSI. In [123] K-means clustering is compared to fixed quantization, where, group subspaces are fixed a priori. Both schemes consider full knowledge of users angle of arrival (AoA) and base station angular spread which in many cases is not a reality. More recently, [124] approximates a lower bound on the expected SINR value for user grouping.

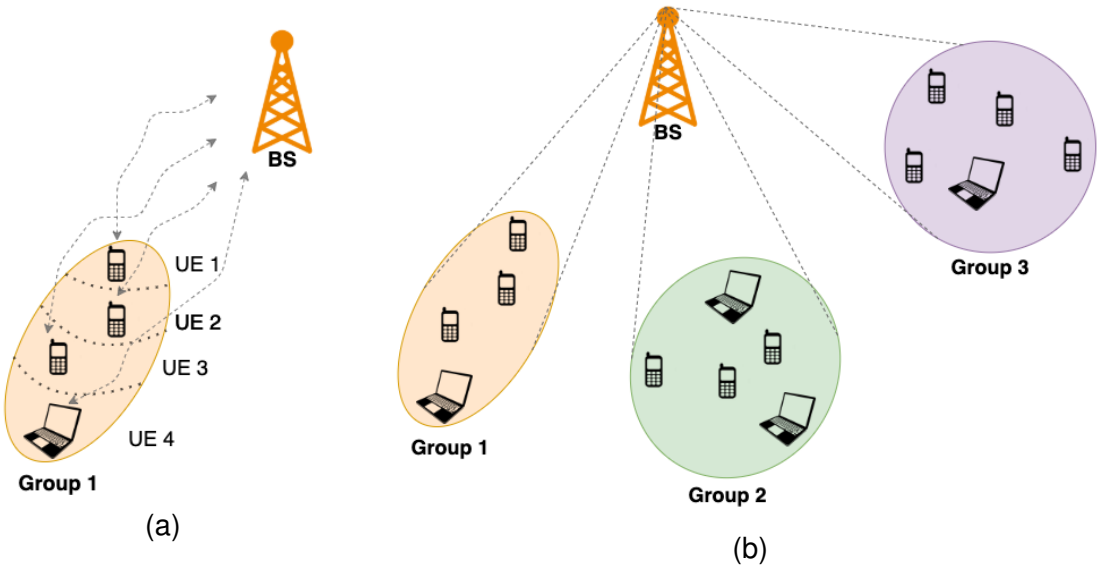


Figure 4.2: Comparison of groups clustered according to (a) channel qualities and (b) spatial location.

In their approach, the authors propose an iterative procedure for obtaining the dominant eigenvectors of the average eigenspaces of users which is then used during clustering.

Comparing the set of all combinations proves to be computationally expensive for a large number of transmitters–receivers. As it stands, classical machine learning approaches can be of extensive help in order to spatially cluster users. In fact, most recent convolutional neural network applications in different fields [110, 113] have proved to be excellent feature extractors. The use of autoencoders proves to be a very efficient data embedding mechanism and led to remarkable results in clustering problems [125]. As previously mentioned, real-world deployed natural language technologies are founded on feature embedding mechanisms. This brings us to question if one could apply similar mechanisms to build more robust and efficient beam selection mechanisms. Natural approaches considering embedding mechanisms either try to solve a simpler problem in a higher dimensional space or try to project the problem into a lower dimension and in which faster optimization algorithms can operate. One way or another, different challenges are posed from the characteristics of the application. As a rule, differently from other fields which naturally contain abundance on data, it is hard to acquire real-world labelled data from the PHY or MAC layer in wireless communication. Moreover, data augmentation techniques as scaling, rotating and clipping are usually not valid for channel state information and signal processing. These are not, however, discouraging obstacles, but interesting challenge which can provide significant insights and gains for the task of user clustering.

Downlink assumes different challenges when compared to uplink. In the former, channel estimates are not immediately available and require broadcasting in order to access relevant users. For channel estimation at the user side, each antenna at the base station must transmit pre-assigned pilots and which are then used at the receiver side to retrieve the channel. A large number of antennas and the wasted radiation power thwart efficient communication. Moreover, when considering FDD transmissions, channel reciprocity does not necessarily hold in real-world scenarios [122]. To cope with that, recent works have

proposed downlink channel estimation based on uplink characteristics [126]. The authors use deep neural networks as a universal function approximator and construct channels-to-channel mapping at different frequencies. Compared to full knowledge over CSI, there was 15% loss in spectral efficiency which proves the effectiveness of the approach in the studied scenario. Despite the valuable results, as suggested by the authors, further investigation is made necessary in order to study various practical considerations.
Similarly, the authors of [127] argue that selecting a subset of the available antennas for transmission plays an important role in saving hardware and energy resources. The joint multicast beamforming and antenna selection problem is approached by a deep neural network trained on channel second-order statistics and predicting a binary vector which represents the activation or deactivation of certain antennas. Data is acquired by an exhaustive search which brings into question the applicability of the solution in massive MIMO scenarios.
A final consideration regarding the use of data-driven solutions in wireless applications relates to considerations between real-world application and simulations. We are yet to describe how to come up with meaningful real-word data which can properly grasp different channel characteristics and their intrinsic meaning, e.g., spatial location, AoA, fading. Moreover, an interesting line of research lies in the mapping between channels with different characteristics, which might give some intuition on how to deploy real-world solutions trained with simulation-based data.

4.4. Summary of Challenges

In this Chapter, we have discussed open challenges regarding the applications of machine learning in real-world massive MIMO communication systems. Three different MIMO related technologies were used to discuss challenges of machine learning when applied to wireless communication. As conclusion, the following research directions are of fundamental importance for robust machine learning solutions in mMIMO

1. Development of efficient methods for real-data acquisition on massive MIMO applications, as well as, transfer learning mechanisms from simulated data to real-world scenarios.
2. Building an asymptotic analysis frameworks to consider black-box solutions;
3. Novel mechanisms to ensure requirements on QoS during training and evaluation;
4. Approaches on scalability issues when considering neural networks;
5. Deep feature-based clustering solutions for the problem of user and antenna grouping.

5. References

[1] C. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, no. 4, pp. 623–666, 1948.

[2] K. P. Murphy, *Machine learning : a probabilistic perspective*. MIT Press, 2012.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25* (F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, eds.), pp. 1097–1105, Curran Associates, Inc., 2012.

[4] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Bell System Technical Journal*, vol. 1, no. 4, pp. 541–551, 1989.

[5] "Globecom 2018 machine learning workshop." https://globecom2018.ieee-globecom.org/workshop/ws-18-mlcomm-machine-learning-communications. Accessed: 2020-08-03.

[6] "Globecom 2019 machine learning workshop." https://globecom2019.ieee-globecom.org/workshop/ws-18-machine-learning-wireless-communications. Accessed: 2020-08-03.

[7] "Globecom 2020 machine learning workshop." https://globecom2020.ieee-globecom.org/workshop/ws-12-open-workshop-machine-learning-communications-ope Accessed: 2020-08-03.

[8] "ICC 2021 machine learning symposium." https://icc2021.ieee-icc.org/sites/icc2021.ieee-icc.org/files/symposia%20CFP/ICC2021-SAC-ML-CFP-V3-Final.pdf. Accessed: 2020-08-03.

[9] "Relevant Papers in machine learning." https://www.comsoc.org/publications/best-readings/machine-learning-communications. Accessed: 2020-08-03.

[10] S. Chen, G. Gibson, and C. Cowan, "Adaptive equalization of finite non-linear channels using multilayer perceptrons," *Signal Processing*, vol. 20, no. 2, pp. 107–119, 1990.

[11] V. N. Vapnik, *Statistical Learning Theory*. Wiley-Interscience, 1998.

[12] H. Shimodaira, "Improving predictive inference under covariate shift by weighting the log-likelihood function," *Journal of Statistical Planning and Inference*, vol. 90, no. 2, pp. 227 – 244, 2000.

[13] C. R. Shelton, "Importance sampling for reinforcement learning with multiple objectives," 2001.

[14] B. Schölkopf, J. Platt, and T. Hofmann, *Dirichlet-Enhanced Spam Filtering based on Biased Samples*, pp. 161–168. 2007.

[15] J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller, and T. M. Vaughan, "Brain–computer interfaces for communication and control," *Clinical Neurophysiology*, vol. 113, no. 6, pp. 767 – 791, 2002.

[16] B. Scholkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA, USA: MIT Press, 2001.

[17] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.

[18] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," in *Advances in Neural Information Processing Systems 20* (J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, eds.), pp. 1177–1184, Curran Associates, Inc., 2008.

[19] A. Rahimi and B. Recht, "Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning," in *Advances in Neural Information Processing Systems 21* (D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, eds.), pp. 1313–1320, Curran Associates, Inc., 2009.

[20] A. Rudi and L. Rosasco, "Generalization properties of learning with random features," 2016.

[21] J. Huang, A. Gretton, K. Borgwardt, B. Schölkopf, and A. J. Smola, "Correcting sample selection bias by unlabeled data," in *Advances in Neural Information Processing Systems 19* (B. Schölkopf, J. C. Platt, and T. Hoffman, eds.), pp. 601–608, MIT Press, 2007.

[22] P. L. Bartlett and S. Mendelson, "Rademacher and gaussian complexities: Risk bounds and structural results," *J. Mach. Learn. Res.*, vol. 3, p. 463–482, Mar. 2003.

[23] T. Zugno, M. Polese, M. Lecci, and M. Zorzi, "Simulation of next-generation cellular networks with ns-3: Open challenges and new directions," in *Proceedings of the 2019 Workshop on Next-Generation Wireless with ns-3*, pp. 38–41, 2019.

[24] M. Agiwal, A. Roy, and N. Saxena, "Next generation 5g wireless networks: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, pp. 1617–1655, 2016.

[25] A. Klautau, N. González-Prelcic, and R. W. Heath, "Lidar data for deep learning-based mmwave beam-selection," *IEEE Wireless Communications Letters*, vol. 8, no. 3, pp. 909–912, 2019.

[26] H. Rutagemwa, K. E. Baddour, C. Brown, and S. Raut, "Evaluation metrics and simulation tools for 5g millimeter-wave networks," *IEEE Wireless Communications*, vol. 25, no. 4, pp. 52–57, 2018.

[27] A. Klautau, P. Batista, N. González-Prelcic, Y. Wang, and R. W. Heath, "5g mimo data for machine learning: Application to beam-selection using deep learning," in *2018 Information Theory and Applications Workshop (ITA)*, pp. 1–9, IEEE, 2018.

[28] I. Trindade, B. V. Boas, and A. Klautau, "Evaluation of simplified methodology for obtaining mmwave mimo channels from ray-tracing simulations," *arXiv preprint arXiv:1908.07126*, 2019.

[29] M. Mezzavilla, M. Zhang, M. Polese, R. Ford, S. Dutta, S. Rangan, and M. Zorzi, "End-to-end simulation of 5g mmwave networks," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 2237–2263, 2018.

[30] S. Choi, J. Song, J. Kim, S. Lim, S. Choi, T. T. Kwon, and S. Bahk, "5g k-simnet: End-to-end performance evaluation of 5g cellular systems," in *2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, pp. 1–6, IEEE, 2019.

[31] M. Zhang, M. Mezzavilla, J. Zhu, S. Rangan, and S. Panwar, "Tcp dynamics over mmwave links," in *2017 IEEE 18th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pp. 1–6, IEEE, 2017.

[32] M. Polese, M. Giordani, A. Roy, S. Goyal, D. Castor, and M. Zorzi, "End-to-end simulation of integrated access and backhaul at mmwaves," in *2018 IEEE 23rd International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, pp. 1–7, IEEE, 2018.

[33] M. K. Müller, F. Ademaj, T. Dittrich, A. Fastenbauer, B. R. Elbal, A. Nabavi, L. Nagel, S. Schwarz, and M. Rupp, "Flexible multi-node simulation of cellular mobile communications: the vienna 5g system level simulator," *EURASIP Journal on Wireless Communications and Networking*, vol. 2018, no. 1, p. 227, 2018.

[34] R. I. Tinini, M. R. P. dos Santos, G. B. Figueiredo, and D. M. Batista, "5gpy: A simpy-based simulator for performance evaluations in 5g hybrid cloud-fog ran architectures," *Simulation Modelling Practice and Theory*, p. 102030, 2019.

[35] S. Pratschner, B. Tahir, L. Marijanovic, M. Mussbah, K. Kirev, R. Nissel, S. Schwarz, and M. Rupp, "Versatile mobile communications simulation: The vienna 5g link level simulator," *EURASIP Journal on Wireless Communications and Networking*, vol. 2018, no. 1, p. 226, 2018.

[36] S. Ghosh, S. Busari, T. Dagiuklas, M. Iqbal, R. Mumtaz, J. Gonzalez, S. Stavrou, and L. Kanaris, "Sdn-sim: Integrating a system-level simulator with a software defined network," *IEEE Communications Standards Magazine*, vol. 4, no. 1, pp. 18–25, 2020.

[37] D. Raca, D. Leahy, C. J. Sreenan, and J. J. Quinlan, "Beyond throughput, the next generation: a 5g dataset with channel and context metrics," in *Proceedings of the 11th ACM Multimedia Systems Conference*, pp. 303–308, 2020.

[38] A. Alkhateeb, "Deepmimo: A generic deep learning dataset for millimeter wave and massive mimo applications," *arXiv preprint arXiv:1902.06435*, 2019.

[39] C. Bouras, G. Diles, A. Gkamas, and A. Zacharopoulos, "Comparison of 4g and 5g network simulators," in *The Fifteenth International Conference on Wireless and Mobile Communications (ICWMC 2019)*, pp. 13–18, 2019.

[40] P. K. Gkonis, P. T. Trakadas, and D. I. Kaklamani, "A comprehensive study on simulation techniques for 5g networks: State of the art results, analysis, and future challenges," *Electronics*, vol. 9, no. 3, p. 468, 2020.

[41] Y. Wang, J. Xu, and L. Jiang, "Challenges of system-level simulations and performance evaluation for 5g wireless networks," *IEEE Access*, vol. 2, pp. 1553–1561, 2014.

[42] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," in *Advances in neural information processing systems*, pp. 2234–2242, 2016.

[43] S. Gurumurthy, R. Kiran Sarvadevabhatla, and R. Venkatesh Babu, "Deligan: Generative adversarial networks for diverse and limited data," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 166–174, 2017.

[44] T. Che, Y. Li, A. P. Jacob, Y. Bengio, and W. Li, "Mode regularized generative adversarial networks," *arXiv preprint arXiv:1612.02136*, 2016.

[45] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," in *Advances in neural information processing systems*, pp. 6626–6637, 2017.

[46] M. S. Sajjadi, O. Bachem, M. Lucic, O. Bousquet, and S. Gelly, "Assessing generative models via precision and recall," in *Advances in Neural Information Processing Systems*, pp. 5228–5237, 2018.

[47] Y. Wu, Y. Burda, R. Salakhutdinov, and R. Grosse, "On the quantitative analysis of decoder-based generative models," *arXiv preprint arXiv:1611.04273*, 2016.

[48] P. Sánchez-Martín, P. M. Olmos, and F. Perez-Cruz, "Improved bigan training with marginal likelihood equalization," *arXiv preprint arXiv:1911.01425*, 2019.

[49] A. Grover, M. Dhar, and S. Ermon, "Flow-gan: Bridging implicit and prescribed learning in generative models," *arXiv preprint arXiv:1705.08868*, vol. 1, 2017.

[50] P. Grnarova, K. Y. Levy, A. Lucchi, N. Perraudin, I. Goodfellow, T. Hofmann, and A. Krause, "A domain agnostic measure for monitoring and evaluating gans," in *Advances in Neural Information Processing Systems*, pp. 12069–12079, 2019.

[51] A. Borji, "Pros and cons of gan evaluation measures," *Computer Vision and Image Understanding*, vol. 179, pp. 41–65, 2019.

[52] L. Theis, A. v. d. Oord, and M. Bethge, "A note on the evaluation of generative models," *arXiv preprint arXiv:1511.01844*, 2015.

[53] W. Jitkrittum, H. Kanagawa, P. Sangkloy, J. Hays, B. Schölkopf, and A. Gretton, "Informative features for model comparison," in *Advances in Neural Information Processing Systems*, pp. 808–819, 2018.

[54] H. Wang, X. Wu, Z. Huang, and E. P. Xing, "High-frequency component helps explain the generalization of convolutional neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8684–8694, 2020.

[55] Z. Wang, Y. Yang, A. Shrivastava, V. Rawal, and Z. Ding, "Towards frequency-based explanation for robust cnn," *arXiv preprint arXiv:2005.03141*, 2020.

[56] A. Radhakrishnan, K. Yang, M. Belkin, and C. Uhler, "Memorization in overparameterized autoencoders," *arXiv preprint arXiv:1810.10333*, 2018.

[57] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.

[58] R. Salakhutdinov and G. Hinton, "Deep boltzmann machines," in *Artificial intelligence and statistics*, pp. 448–455, 2009.

[59] R. Chellappa and S. Chatterjee, "Classification of textures using gaussian markov random fields," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 33, no. 4, pp. 959–963, 1985.

[60] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural computation*, vol. 14, no. 8, pp. 1771–1800, 2002.

[61] A. Hyvärinen, "Estimation of non-normalized statistical models by score matching," *Journal of Machine Learning Research*, vol. 6, no. Apr, pp. 695–709, 2005.

[62] J. Chen, C. Lu, B. Chenli, J. Zhu, and T. Tian, "Vflow: More expressive generative flows with variational data augmentation," *arXiv preprint arXiv:2002.09741*, 2020.

[63] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[64] Y. Burda, R. Grosse, and R. Salakhutdinov, "Importance weighted autoencoders," *arXiv preprint arXiv:1509.00519*, 2015.

[65] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," *arXiv preprint arXiv:1401.4082*, 2014.

[66] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, pp. 2672–2680, 2014.

[67] S. Nowozin, B. Cseke, and R. Tomioka, "f-gan: Training generative neural samplers using variational divergence minimization," in *Advances in neural information processing systems*, pp. 271–279, 2016.

[68] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein gan," *arXiv preprint arXiv:1701.07875*, 2017.

[69] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," in *Advances in neural information processing systems*, pp. 5767–5777, 2017.

[70] A. B. Dieng, F. J. Ruiz, D. M. Blei, and M. K. Titsias, "Prescribed generative adversarial networks," *arXiv preprint arXiv:1910.04302*, 2019.

[71] S. Sagawa, A. Raghunathan, P. W. Koh, and P. Liang, "An investigation of why overparameterization exacerbates spurious correlations," *arXiv preprint arXiv:2005.04345*, 2020.

[72] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

[73] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255, Ieee, 2009.

[74] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[75] R. Zhang, P. Isola, and A. A. Efros, "Colorful image colorization," in *European conference on computer vision*, pp. 649–666, Springer, 2016.

[76] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440, 2015.

[77] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1125–1134, 2017.

[78] S. Santurkar, L. Schmidt, and A. Madry, "A classification-based study of covariate shift in gan distributions," in *International Conference on Machine Learning*, pp. 4480–4489, 2018.

[79] J. Yang, A. Kannan, D. Batra, and D. Parikh, "Lr-gan: Layered recursive generative adversarial networks for image generation," *arXiv preprint arXiv:1703.01560*, 2017.

[80] T. Lesort, F. Bordes, J.-F. Goudou, and D. Filliat, "Evaluation of generative networks through their data augmentation capacity," *URL https://openreview.net/forum?id=HJ1HFlZAb*, 2018.

[81] R. M. Neal, "Annealed importance sampling," *Statistics and computing*, vol. 11, no. 2, pp. 125–139, 2001.

[82] J. Donahue, P. Krähenbühl, and T. Darrell, "Adversarial feature learning," *arXiv preprint arXiv:1605.09782*, 2016.

[83] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *The Journal of Machine Learning Research*, vol. 13, no. 1, pp. 723–773, 2012.

[84] S. Arora and Y. Zhang, "Do gans actually learn the distribution? an empirical study," *arXiv preprint arXiv:1706.08224*, 2017.

[85] I. O. Tolstikhin, S. Gelly, O. Bousquet, C.-J. Simon-Gabriel, and B. Schölkopf, "Adagan: Boosting generative models," in *Advances in Neural Information Processing Systems*, pp. 5424–5433, 2017.

[86] E. Richardson and Y. Weiss, "On gans and gmms," in *Advances in Neural Information Processing Systems*, pp. 5847–5858, 2018.

[87] S. Huang, A. Makhzani, Y. Cao, and R. Grosse, "Evaluating lossy compression rates of deep generative models," *International Conference on Machine Learning*, 2019.

[88] M. Belkin, S. Ma, and S. Mandal, "To understand deep learning we need to understand kernel learning," *arXiv preprint arXiv:1802.01396*, 2018.

[89] E. Björnson, L. Sanguinetti, J. Hoydis, and M. Debbah, "Designing multi-user mimo for energy efficiency: When is massive mimo the answer?," in *2014 IEEE wireless communications and networking conference (WCNC)*, pp. 242–247, IEEE, 2014.

[90] E. Björnson, L. Sanguinetti, H. Wymeersch, J. Hoydis, and T. L. Marzetta, "Massive mimo is a reality—what is next?: Five promising research directions for antenna arrays," *Digital Signal Processing*, vol. 94, pp. 3–20, 2019.

[91] B. Matthiesen, A. Zappone, K.-L. Besser, E. A. Jorswieck, and M. Debbah, "A globally optimal energy-efficient power control framework and its efficient implementation in wireless interference networks," *arXiv preprint arXiv:1812.06920*, 2018.

[92] A. Zappone, L. Sanguinetti, and M. Debbah, "User association and load balancing for massive mimo through deep learning," in *2018 52nd Asilomar Conference on Signals, Systems, and Computers*, pp. 1262–1266, IEEE, 2018.

[93] F. A. Aoudia and J. Hoydis, "Model-free training of end-to-end communication systems," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 11, pp. 2503–2516, 2019.

[94] M. Khani, M. Alizadeh, J. Hoydis, and P. Fleming, "Adaptive neural signal detection for massive mimo," *IEEE Transactions on Wireless Communications*, 2020.

[95] W. Cui, K. Shen, and W. Yu, "Spatial deep learning for wireless scheduling," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1248–1261, 2019.

[96] H. Sun, X. Chen, Q. Shi, M. Hong, X. Fu, and N. D. Sidiropoulos, "Learning to optimize: Training deep neural networks for wireless resource management," in *2017 IEEE 18th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pp. 1–6, IEEE, 2017.

[97] T. Van Chien, E. Bjornson, and E. G. Larsson, "Sum spectral efficiency maximization in massive mimo systems: Benefits from deep learning," in *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, pp. 1–6, IEEE Deep Learning for Massive MIMO CSI Feedback, 2019.

[98] H. Q. Ngo, A. Ashikhmin, H. Yang, E. G. Larsson, and T. L. Marzetta, "Cell-free massive mimo: Uniformly great service for everyone," in *2015 IEEE 16th international workshop on signal processing advances in wireless communications (SPAWC)*, pp. 201–205, IEEE, 2015.

[99] H. Q. Ngo, A. Ashikhmin, H. Yang, E. G. Larsson, and T. L. Marzetta, "Cell-free massive mimo versus small cells," *IEEE Transactions on Wireless Communications*, vol. 16, no. 3, pp. 1834–1850, 2017.

[100] M. Bashar, K. Cumanan, A. G. Burr, M. Debbah, and H. Q. Ngo, "On the uplink max–min sinr of cell-free massive mimo systems," *IEEE Transactions on Wireless Communications*, vol. 18, no. 4, pp. 2021–2036, 2019.

[101] T. T. Vu, D. T. Ngo, N. H. Tran, H. Q. Ngo, M. N. Dao, and R. H. Middleton, "Cell-free massive mimo for wireless federated learning," *IEEE Transactions on Wireless Communications*, 2020.

[102] S. Buzzi and C. D'Andrea, "Cell-free massive mimo: User-centric approach," *IEEE Wireless Communications Letters*, vol. 6, no. 6, pp. 706–709, 2017.

[103] M. Attarifar, A. Abbasfar, and A. Lozano, "Random vs structured pilot assignment in cell-free massive mimo wireless networks," in *2018 IEEE International Conference on Communications Workshops (ICC Workshops)*, pp. 1–6, IEEE, 2018.

[104] F. Riera-Palou, G. Femenias, A. G. Armada, and A. Pérez-Neira, "Clustered cell-free massive mimo," in *2018 IEEE Globecom Workshops (GC Wkshps)*, pp. 1–6, IEEE, 2018.

[105] E. Nayebi, A. Ashikhmin, T. L. Marzetta, H. Yang, and B. D. Rao, "Precoding and power optimization in cell-free massive mimo systems," *IEEE Transactions on Wireless Communications*, vol. 16, no. 7, pp. 4445–4459, 2017.

[106] S. Dörner, S. Cammerer, J. Hoydis, and S. Ten Brink, "Deep learning based communication over the air," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 132–143, 2017.

[107] T. Van Chien, T. N. Canh, E. Björnson, and E. G. Larsson, "Power control in cellular massive mimo with varying user activity: A deep learning solution," *IEEE Transactions on Wireless Communications*, 2020.

[108] A. Klautau, N. González-Prelcic, and R. W. Heath, "Lidar data for deep learning-based mmwave beam-selection," *IEEE Wireless Communications Letters*, vol. 8, no. 3, pp. 909–912, 2019.

[109] Y. Goldberg and O. Levy, "word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method," *arXiv preprint arXiv:1402.3722*, 2014.

[110] F. Tan, S. Feng, and V. Ordonez, "Text2scene: Generating compositional scenes from textual descriptions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[111] Y. Zheng, Y. Li, and S. Wang, "Intention oriented image captions with guiding objects," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[112] O. Vinyals, C. Blundell, T. Lillicrap, k. kavukcuoglu, and D. Wierstra, "Matching networks for one shot learning," in *Advances in Neural Information Processing Systems 29* (D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, eds.), pp. 3630–3638, Curran Associates, Inc., 2016.

[113] K. Li, Z. Wu, K.-C. Peng, J. Ernst, and Y. Fu, "Tell me where to look: Guided attention inference network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[114] B. Carter, J. Mueller, S. Jain, and D. Gifford, "What made you do this? understanding black-box decisions with sufficient input subsets," in *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 567–576, 2019.

[115] H. Fukui, T. Hirakawa, T. Yamashita, and H. Fujiyoshi, "Attention branch network: Learning of attention mechanism for visual explanation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 10705–10714, 2019.

[116] H. Qiang, G. Feifei, Z. Hao, J. Shi, and L. G. Ye, "Deep learning for mimo channel estimation: Interpretation, performance, and comparison," *arXiv preprint arXiv:1911.01918*, 2019.

[117] M. S. Sim, Y.-G. Lim, S. H. Park, L. Dai, and C.-B. Chae, "Deep learning-based mmwave beam selection for 5g nr/6g with sub-6 ghz channel information: Algorithms and prototype validation," *IEEE Access*, vol. 8, pp. 51634–51646, 2020.

[118] H. Huang, W. Xia, J. Xiong, J. Yang, G. Zheng, and X. Zhu, "Unsupervised learning-based fast beamforming design for downlink mimo," *IEEE Access*, vol. 7, pp. 7599–7605, 2018.

[119] N. González-Prelcic, A. Ali, V. Va, and R. W. Heath, "Millimeter-wave communication with out-of-band information," *IEEE Communications Magazine*, vol. 55, no. 12, pp. 140–146, 2017.

[120] M. Saideh, Y. Alsaba, I. Dayoub, and M. Berbineau, "Joint interference cancellation for multi-carrier modulation-based non-orthogonal multiple access," *IEEE Communications Letters*, vol. 23, no. 11, pp. 2114–2117, 2019.

[121] M. A. Sedaghat and R. R. Müller, "On user pairing in uplink noma," *IEEE Transactions on Wireless Communications*, vol. 17, no. 5, pp. 3474–3486, 2018.

[122] A. Adhikary, J. Nam, J.-Y. Ahn, and G. Caire, "Joint spatial division and multiplexing—the large-scale array regime," *IEEE transactions on information theory*, vol. 59, no. 10, pp. 6441–6463, 2013.

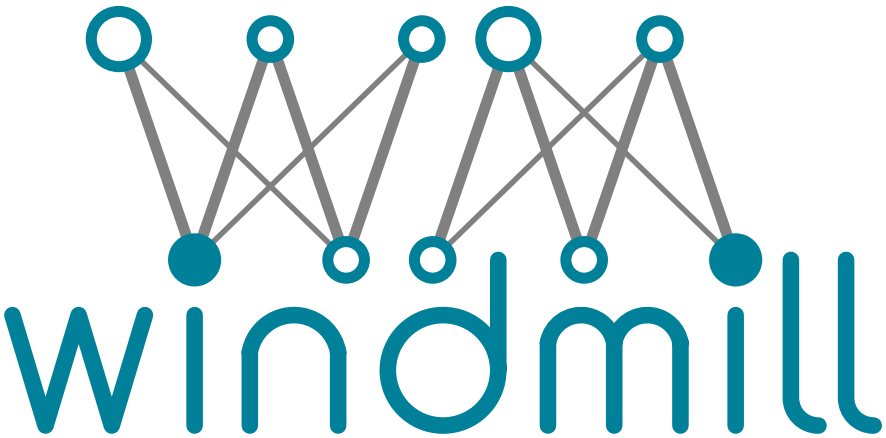
[123] J. Nam, A. Adhikary, J.-Y. Ahn, and G. Caire, "Joint spatial division and multiplexing: Opportunistic beamforming, user grouping and simplified downlink scheduling," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 5, pp. 876–890, 2014.

[124] T.-K. Kim, "User scheduling and grouping in massive mimo broadcast channels with heterogeneous users," *Journal of Communications and Networks*, vol. 21, no. 4, pp. 385–394, 2019.

[125] E. Min, X. Guo, Q. Liu, G. Zhang, J. Cui, and J. Long, "A survey of clustering with deep learning: From the perspective of network architecture," *IEEE Access*, vol. 6, pp. 39501–39514, 2018.

[126] M. Arnold, S. Dörner, S. Cammerer, S. Yan, J. Hoydis, and S. t. Brink, "Enabling fdd massive mimo through deep learning-based channel prediction," *arXiv preprint arXiv:1901.03664*, 2019.

[127] M. S. Ibrahim, A. S. Zamzam, X. Fu, and N. D. Sidiropoulos, "Learning-based antenna selection for multicasting," in *2018 IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pp. 1–5, IEEE, 2018.



www.windmill-itn.eu