RICE UNIVERSITY

Annealed Langevin Dynamics for Massive MIMO Communications

By

Nicolas Zilberstein

A THESIS SUBMITTED
IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE

Master of Science

APPROVED, THESIS COMMITTEE

Santiago Segarra
Santiago Segarra (Nov 15, 2023 12:03 EST)

Santiago Segarra (Chair)
Assistant Professor of Electrical and
Computer Engineering

Ashutosh Sabharwal
Ernest Dell Butcher Professor of Electrical
and Computer Engineering

Cesar A Uribe
Cesar A Uribe (Nov 17, 2023 10:39 CST)

Cesar Uribe
Assistant Professor of Electrical and
Computer Engineering

HOUSTON, TEXAS

November 2023

RICE UNIVERSITY

Annealed Langevin Dynamics for Massive Mimo Communications

by

Nicolas Zilberstein

A Thesis Submitted
in Partial Fulfillment of the
Requirements for the Degree

Master of Science

Approved, Thesis Committee:

Santiago Segarra, Chair
Assistant Professor of Electrical and
Computer Engineering

Cesar Uribe
Assistant Professor of Electrical and
Computer Engineering

Ashutosh Sabharwal
Ernest Dell Butcher Professor of
Electrical and Computer Engineering

Houston, Texas

November, 2023

ABSTRACT

Annealed Langevin Dynamics for Massive Mimo Communications

by

Nicolas Zilberstein

Solving the optimal data detection problem in multiple-input multiple-output (MIMO) systems is known to be NP-hard. Moreover, the difficulty is exacerbated when the channel state information is unavailable. In this work we propose a MIMO detector for the two scenarios, namely when the CSI is known and when it is unknown. First, for the case of perfect CSI, we proposed a MIMO detector based on an annealed version of Langevin dynamics. More precisely, we define a stochastic dynamical process whose stationary distribution coincides with the posterior distribution of the data given our observations. This allows us to approximate the maximum a posteriori estimator of the transmitted symbols by sampling from the proposed Langevin dynamic. We carefully craft this stochastic dynamic by gradually adding a sequence of noise with decreasing variance to the trajectories, which ensures that the estimated symbols belong to a pre-specified discrete constellation. Second, for the case of unknown CSI, we propose a joint data detection and channel estimation solution, where we define an annealed Langevin diffusion whose stationary distribution is the joint posterior of the channels and data given noisy observations.

Contents

Abstract ii

List of Illustrations v

List of Tables viii

1 Introduction 1

1.1 Data detection with known CSI 2

1.2 Data detection with unknown CSI 3

1.3 Outline 4

2 Background 5

2.1 Langevin diffusion 5

2.2 Score-based generative modeling 7

3 Massive MIMO detection based on annealed Langevin dynamics 10

3.1 System model 10

3.2 Related work 12

3.2.1 Classical MIMO detectors. 13

3.2.2 Learning-based MIMO detectors. 15

3.3 Detection by sampling from the posterior distribution 16

3.3.1 Annealing process. 18

3.3.2 Synthetic noise construction. 19

3.3.3 Score functions. 20

3.3.4 Langevin-based MIMO detector algorithm. 21

3.4 Unfolding of the Langevin detector 24
3.4.1 Background in algorithm unfolding 25
3.4.2 Detection by sampling from the posterior distribution using a learned score network. 26
3.5 Results 28
3.5.1 Analysis of Langevin method 30
3.5.2 Comparison with other methods 33
3.5.3 Real-world data and measured channel 36
3.5.4 Performance for multiple modulation schemes simultaneously . 37
3.5.5 Imperfect CSI 39
3.5.6 Line-of-sight (LOS) and non-LOS (NLOS) channels 41
3.5.7 Robustness to noise variance uncertainty 41

4 Joint Massive MIMO data detection and channel estimation based on Lagevin diffusion 43
4.1 System model and problem formulation 43
4.2 Joint sampling based on Langevin diffusion. 45
4.2.1 Score functions. 46
4.2.2 Langevin-based joint MIMO data detection and channel estimation algorithm. 48
4.3 Numerical Experiments 50

5 Conclusions 54

Bibliography 56

Illustrations

1.1 Scheme of the annealed process. We consider a QPSK constellation and at each level, we add Gaussian noise. The variance of the noise decreases at higher levels. In the last level L, the Gaussian is very sharp around each symbol, mimicking our true discrete prior over the constellation. 3

3.1 Scheme of our proposed unfolded Langevin detector. The input to architecture is the SVD of $\mathbf{H}$ and the observation $\mathbf{y}$ and the estimated symbol is computed by running sequentially the Langevin dynamic for each of the L levels, represented by blue. Inside each level l, we run (3.21) for T iterations, where the score network $\mathbf{s}_l(.;\boldsymbol{\Theta})$ is the same for all the interatios in the level. This is represented by green, and the output $\tilde{\boldsymbol{\chi}}_{T,l}$ is the initial value for the next level $l-1$. 26

3.2 Performance analysis of our proposed method evaluated in a Kronecker correlated channel model as in (3.24). (a) SER as a function of SNR for our Langevin method for noise levels $L \in \{5, 10, 15, 20\}$. (b) SER as a function of SNR for our Langevin method for $M \in \{1, 5, 10, 20\}$ numbers of trajectories. (c) SER as a function of SNR for two different temperature parameters τ. (d) Numerical convergence of our proposed detector, in terms of SER as a function of iterations. 31

3.3 SER as a function of SNR for different detection methods evaluated in (a) Rayleigh channel model and a modulation 16-QAM. (b) Kronecker correlated channel model as in (3.24) and a modulation 16-QAM. (c) 3GPP channel model and a modulation 16-QAM. (d) 3GPP channel model and a modulation 64-QAM. 34

3.4 (a) SER as a function of SNR for our Langevin method using the real channel defined in Section 3.5.3 and with synthetic data. The modulation is 4-QAM. (b) Histogram of the number of errors per subcarrier and OFDM symbol for each of the 200 time and frame and 8 users. (c)-(d) Symbol's estimation for two users and 200 different observations. We consider one OFDM symbol and one subcarrier for both users, and send the same symbol in the 200 time frames. 38

3.5 Performance analysis of our proposed method evaluated in a Kronecker correlated channel model as in (3.24). (a) SER as a function of SNR for our Langevin method for users transmitting with two 16-QAM and 64-QAM simultaneously. (b) Comparison of our proposed MIMO detectors with other baseline detector when there is imperfect channel state information. We consider SNR = 16dB and $\sigma_e \in \{0.002, 0.02, 0.034, 0.06, 0.1\}$. (c) Comparison of our proposed MIMO detector with the MMSE and ML detectors for channels that belong to the Rician and Rayleigh models. (d) SER as a function of SNR when there is noise variance uncertainty. 40

4.1 Performance analysis of our proposed method as a function of SNR for a 3GPP channel model. NMSE for a fixed number of pilot symbols $P = 30$ and different number of data symbols, $D = \{10, 20, 30, 40, 50, 70, 100\}$ 51

4.2 Performance analysis of our proposed method as a function of SNR for a 3GPP channel model. SER for a fixed number of pilot symbols $P = 30$ and different number of data symbols, $D = \{10, 20, 30, 40, 50, 70, 100\}$ 52
4.3 Comparison with different baseline methods, for $P = 30$ and $D = 50$. 53

Tables

3.1 Complexity of Langevin and the other baseline detectors. We observe that our proposed detector has a polynomial complexity much like the other detectors. 25

Chapter 1

Introduction

Massive multiple-input multiple-output (MIMO) is a key technology for the development of modern and future wireless communications [1], [2]. It is expected to play a key role in moving from the fifth to the sixth generation of cellular communications by achieving high data rates and spectral efficiency [3]. In massive MIMO systems, base stations are equipped with a large number of antennas, enabling them to handle several users simultaneously on the same time-frequency resource. Although this is a key enabler for achieving great throughput and spectral efficiency, it comes at the cost of drastically increasing the complexity of the receiver. In this thesis, we focus on the MIMO data detection problem.

Exact MIMO detection is NP-hard [4]. With N_u users and κ-symbol modulation, the maximum likelihood estimator's decoding complexity is exponential $\mathcal{O}(\kappa^{N_u})$, rendering it infeasible for sizable systems. Additionally, it relies on perfect channel state information (CSI), usually unavailable. Hence, an efficient channel estimation method is crucial, but it is challenging in massive MIMO due to the high dimensionality of the system and the constraint on the number of pilot signals [5]. As a consequence, a joint optimization approach (still NP-hard) to estimate both the data and the channel is necessary to tackle these issues in high-dimensional. Therefore, this thesis focuses on the following scenarios: 1) massive MIMO data detection when the CSI is available, and 2) massive MIMO data detection when the CSI is unavailable, which yields a joint data detection and channel estimation problem.

1.1 Data detection with known CSI

Several approximate solutions for symbol detection have been proposed both in the classical, and more recently, in the machine learning literature. For example, two of the simplest detectors are zero-forcing (ZF) and minimum mean squared error (MMSE) [6]. Although both (linear) detectors have low complexity and perform well on small systems, their performance degrades significantly for larger systems and higher modulation schemes [7]. In the context of machine learning, several massive MIMO symbol detectors have been derived. They have shown promising results, with a good balance between performance and complexity. They can be roughly classified into data-driven methods, like DetNet [8], and model-driven methods, like RE-MIMO [9], OAMPNet [10], and hyperMIMO [11, 12]. The main issue with all of the learning-based detectors is their dependency on a particular configuration of the communication system as they are trained for a specific modulation scheme and, except RE-MIMO, for a given system size.

Overall, both classical and learning-based detectors follow an iterative approach that splits the problem into two steps, where the optimization is done in a continuous space followed by a projection step. Hence, the motivation for this first part of the thesis can be synthesized by the following question:

Can we incorporate the prior information about the discrete nature of the symbols in the optimization problem?

In essence, we seek a detector that solves the optimization problem directly by efficiently searching in the discrete search space. To tackle this, we propose to use an annealed version of *Langevin dynamics* – a low complexity MCMC-based sampler to generate samples from the posterior. At a high level, we propose to approximate the

distribution of the symbols by a sequence of Gaussian noise with mean at each symbol and decreasing variance, thus concentrating around delta distributions at each symbol; see Fig. 1.1.

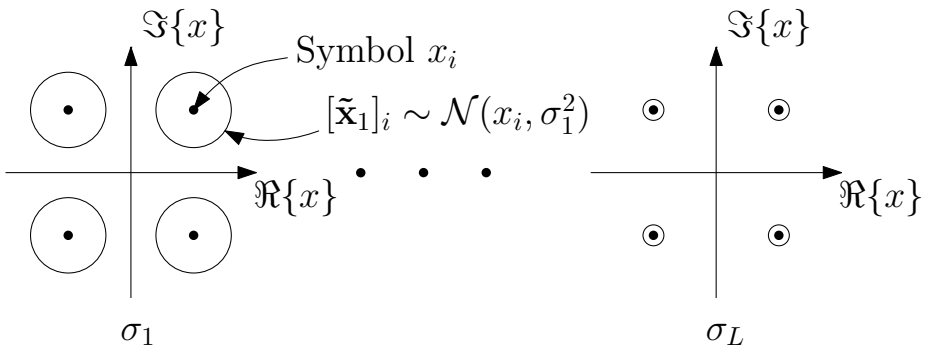


Figure 1.1 : Scheme of the annealed process. We consider a QPSK constellation and at each level, we add Gaussian noise. The variance of the noise decreases at higher levels. In the last level L, the Gaussian is very sharp around each symbol, mimicking our true discrete prior over the constellation.

1.2 Data detection with unknown CSI

Different approximate solutions based on gradient descent have been proposed for single-input multiple-output systems [13]. Recently, new methods have been proposed for massive MIMO. In [14], the authors proposed an algorithm that iteratively solves a relaxed version of the maximum a posteriori joint channel estimation and data detection (MAP-JED) problem that exploits the sparsity of the channel, while in [15] a solution based on deep neural networks, and more specifically, algorithmic unfolding, was proposed.

Therefore, in this second part of the thesis we seek to answer the following question:

Can we derive a method that jointly estimates the data and channels while leveraging prior information of both variables?

In this context, we leverage again an annealed version of *Langevin dynamics*, where we propose a solution for this problem based on score-based generative modeling, a powerful technique that allows us to include prior information through a learned neural network. To achieve this, we construct a diffusion process that models the joint distribution of the channels and symbols given noisy observations and then samples it by running the reverse diffusion.

1.3 Outline

We will cover a brief introduction of Langevin diffusion processes and score-based methods in Chapter 2. In Chapter 3 we present our proposed massive MIMO detector based on annealed Langevin dynamics. Then, in Chapter 4 we present the joint massive MIMO data detection and channel estimation. Concluding remarks are made in Chapter 5.

Chapter 2

Background

2.1 Langevin diffusion

In general, a diffusion process is a continuous-time Markov process on the variable $\mathbf{x} \in \mathbb{R}^d$ that solves the Ito equation

$$\mathrm{d}\mathbf{x}_t = \mathbf{f}(\mathbf{x}_t, t)\mathrm{d}t + \mathbf{g}(\mathbf{x}_t, t)\mathrm{d}\mathbf{w}_t, \tag{2.1}$$

where $\mathbf{w}$ is a standard d-dimensional Brownian motion, $\mathbf{f}(\mathbf{x}_t, t)$ and $\mathbf{g}(\mathbf{x}_t, t)$ are the drift and diffusion term respectively, which are assumed to be Lipschitz continuous for all t. The random variable $\mathbf{x}_t$ is described by the probability density function $p(\mathbf{x}_t, t)$, which is obtained as the solution of a partial differential equation known as *Fokker-Planck*.

$$\frac{\partial \rho}{\partial t} = \nabla \cdot \left(-\mathbf{f}(\mathbf{x}_t, t)p(\mathbf{x}_t, t) + \frac{1}{2}\nabla \cdot (\boldsymbol{g}(\mathbf{x}_t, t)p(\mathbf{x}_t, t)) \right). \tag{2.2}$$

Under the Lipschitz assumptions of the drift and diffuion terms, the solution of (2.1) is unique [16, Theorem 5.2.1].

The Langevin diffusion is a particular type of diffusion process, which is obtained when $f(\mathbf{x}_t, t) = -\nabla U(\mathbf{x}_t)$ and $g(t) = \sqrt{2\tau}$ [17].

$$\mathrm{d}\mathbf{x}_t = -\nabla U(\mathbf{x}_t) + \sqrt{2\tau}\mathrm{d}\mathbf{w}_t, \tag{2.3}$$

where $\mathbf{W}$ is a standard d-dimensional Brownian motion. Here, $U \in \mathcal{C}^2(\mathbb{R}^d)$ is referred to as the potential, and τ is a temperature parameter. The dynamics described

by (2.3) is a particle's movement obeying Newton's second law, influenced by the force $\nabla U(\mathbf{x}_t)$ and interacting with a thermal reservoir through random fluctuations (Brownian motion) representing collisions with other particles. Subject to mild conditions [17], it can be established that the invariant distribution of this continuous-time process is

$$\pi(\mathbf{x}) \propto \exp(-\tau^{-1}U(\mathbf{x})) \tag{2.4}$$

Notice that this invariant distribution lies in the kernel of (2.2) Specifically, for a desired target distribution $p(\mathbf{x})$ from which we aim to generate samples $\mathbf{x} \in \mathbb{R}^d$, with τ fixed at 1, defining $U(\mathbf{x}) = -\log p(\mathbf{x})$ allows us to utilize(2.3) to construct an MCMC sampler, where

$$\pi(\mathbf{x}) = \frac{1}{Z}\, p(\mathbf{x}). \tag{2.5}$$

This leads to an MCMC algorithm [18, 19] known as the unadjusted Langevin algorithm. It is derived through the Euler-Maruyama discretization of (2.3) and is described by the following discrete-time equation.

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \epsilon \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t) + \sqrt{2\epsilon}\, \mathbf{w}_t, \tag{2.6}$$

where $p(\mathbf{x})$ is the target distribution from which we want to generate samples $\mathbf{x} \in \mathbb{R}^N$ and $\mathbf{w}_t \sim \mathcal{N}(0, \mathbf{I}_N)$. The dynamic in (2.6) explores the domain of the target distribution by moving in the direction of the gradient of the logarithm of the target density $\nabla_{\mathbf{x}} \log p(\mathbf{x})$, known as the *score function*. It is essentially a combination of a stochastic gradient ascent in the direction of the score function with a noise term, which prevents the method from collapsing to a local maxima. Under some regularity conditions [20], the distribution of $\mathbf{x}_T$ converges to $p(\mathbf{x})$ when $\epsilon \to 0$ and $T \to \infty$, in which case $\mathbf{x}_T$ becomes an exact sample from $p(\mathbf{x})$. In practice, neither $\epsilon \to 0$ nor $T \to$

∞, so a Metropolis-Hastings acceptance/rejection step is used to ensure convergence, leading to the so-called Metropolis-adjusted Langevin algorithm (MALA) [19]. An alternative, proposed in [20], is a time-inhomogeneous variant of (2.6), i.e., defining a variable step size ϵ_t. Thus, when ϵ_t decreases to zero for large t, the error becomes negligible and the acceptance/rejection step can be omitted. Although this result is asymptotic, also non-asymptotic convergence results in the 2-Wasserstein distance have been obtained when the target distribution is log-concave and smooth (the potential has Lipschitz continuous gradient and is strongly convex) [21].

Temperature parameter. The discretization of the Langevin dynamic in (2.3) can be rewritten for a temperature parameter $\tau \neq 1$, that adds a new degree of freedom:

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \epsilon \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t) + \sqrt{2\epsilon\tau}\, \mathbf{w}_t. \tag{2.7}$$

The behaviour of this dynamic is the same as in (2.6), but the stationary distribution is now $p(\mathbf{x})^{1/\tau}$; see [17, Chapter 6] for details. This additional degree of freedom provides a trade-off between mixing time and fidelity. Indeed, even though the stationary distribution is modified, when $\tau \neq 1$ the mixing time of the dynamic can be improved. This is especially helpful for multimodal distributions.

2.2 Score-based generative modeling

It should be noted that the *only* requirement for sampling from $p(\mathbf{X})$ using this procedure is knowing the score function, which is unknown in general. Hence, we need to estimate the score $\nabla \log p(\mathbf{x})$. To do so, we propose to parameterize the score with a neural network $\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x})$, known a *score network*. Then, the goal is to train the

score network by minimizing the following loss function [22]

$$\mathcal{L}_{ESM}(\boldsymbol{\theta}) = \mathbb{E}_{p(\mathbf{x})}\left[||\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}) - \nabla \log p(\mathbf{x})||_2^2\right]. \tag{2.8}$$

This formulation is known as explicit score matching. Given a training dataset $\{\mathbf{x}_i\}_{i=1}^n$ drawn from the distribution $p(\mathbf{x})$, we can approximate the loss function by the empirical mean and minimize it

$$\mathcal{L}(\boldsymbol{\theta}) \approx \frac{1}{n}\sum_{i=1}^{n} ||\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_i) - \nabla \log p(\mathbf{x}_i)||_2^2 \tag{2.9}$$

However, this problem is intractable because we do not have access to the true score $\nabla \log p(\mathbf{x})$. To circumvent this, we consider denoising score matching [23]. Given the training data, we define its associated perturbed value as $\tilde{\mathbf{x}}_i$ by adding some noise. This defines a distribution over the corrupted data $q(\tilde{\mathbf{x}}|\mathbf{x})$. Then, we define the following loss function.

$$\mathcal{L}_{DSM}(\boldsymbol{\theta}) = \mathbb{E}_{p(\mathbf{x})q(\tilde{\mathbf{x}}|\mathbf{x})}\left[\left|\left|\mathbf{s}_{\boldsymbol{\theta}}(\tilde{\mathbf{x}}) - \frac{\partial \log q(\tilde{\mathbf{x}}|\mathbf{x})}{\partial \tilde{\mathbf{x}}}\right|\right|_2^2\right]. \tag{2.10}$$

It is shown in [23] that the losses $\mathcal{L}_{DSM}(\boldsymbol{\theta})$ and $nccalL_\{ESM\}(nbbtheta)$ are equivalent from an optimization standpoint. In general, the perturbation is additive and Gaussian with zero mean and diagonal covariance matrix $\sigma_l\mathbf{I}$. This yields a tractable expression for the conditional score

$$\frac{\partial \log q(\tilde{\mathbf{x}}|\mathbf{x})}{\partial \tilde{\mathbf{x}}} = -\frac{\tilde{\mathbf{x}} - \mathbf{x}}{\sigma_l^2} \tag{2.11}$$

Finally, the work in [24] proposed to perturb the data at different scales with Gaussian kernels of variance $\{\sigma_l\}_{l=1}^L$ associated with each scale l. Then, they estimate a joint score network $\mathbf{s}_\theta(\mathbf{X}, \sigma)$ by minimizing a weighted version of the loss in (**??**) given by

$$\mathcal{L}_{DSM}(\boldsymbol{\theta}, \{\sigma_l\}_{l=1}^L) = \frac{1}{L}\sum_{l=1}^{L} \lambda(\sigma_l)\mathbb{E}_{p(\mathbf{x})q_{\sigma_l}(\tilde{\mathbf{x}}|\mathbf{x})}\left[\left|\left|\mathbf{s}_\theta(\tilde{\mathbf{x}}, \sigma_l) + \frac{\tilde{\mathbf{x}} - \mathbf{x}}{\sigma_l^2}\right|\right|_2^2\right]. \tag{2.12}$$

where $\lambda(\sigma_l)$ is a pre-defined weight depending on σ_l. After training, we can replace $\nabla_{\mathbf{X}_k} \log p(\mathbf{X}_k)$ in (2.6) by $\mathbf{s}_\theta(\mathbf{X}, \sigma)$ and generate samples from the target distribution $p(\mathbf{X})$.

Related work of diffusion models in the context of inverse problem. Diffusion models, known for their versatility, have excelled across diverse applications [25]. These models transform training data into white Gaussian noise (the forward process) and then run the reverse transformation (the generative process) to generate new data. In [24, 26], an *annealed* Langevin dynamic is used in the context of generative modeling for images. Assuming an unknown distribution of the images, the authors propose to parameterize the score function as a neural network and use the annealed Langevin dynamic to sample from the underlying probability distribution. Leveraging their ability to generate high-dimensional data from unknown probability distributions, diffusion models have been used a generative prior in various inverse problems, achieving state-of-the-art results. For instance, in the context of image processing, they have been used for denoising and image inpainting [27, 28]. In wireless technology, they have been used to derive a massive MIMO detector based on Langevin diffusion [29, 30], and for channel estimation [31], both of which have shown remarkable results. They have been also used for blind inverse problems, where the forward operator is uknown [32, 33].

Chapter 3

Massive MIMO detection based on annealed Langevin dynamics

In this chapter, we address the problem of data detection in massive MIMO systems. That is, we seek to infer the transmitted data based on noisy observations and the channel state information. The direct solution, namely maximum-likelihood (ML), is NP-hard. In this context, I present my proposed detector based on the Langevin diffusion introduced in Section 2.1. The organization of this chapter is as follows: in Section 3.1, we present the system model and formulate the MIMO detection problem. Given this formulation of the problem, we give an overview of previous detectors. In Section 3.3, we provide a detailed description of our proposed detector. In Section 3.4, we describe the proposed unfolded detector. We illustrate results through numerical experimentation, demonstrating the superior performance of our detector in comparison with other benchmark methods in Section 3.5

3.1 System model

We consider a communication channel with N_u single-antenna transmitters or users and a receiving base station with N_r antennas. The forward model for this MIMO system is given by

$$\bar{\mathbf{y}} = \bar{\mathbf{H}}\bar{\mathbf{x}} + \bar{\mathbf{z}}, \tag{3.1}$$

where $\bar{\mathbf{H}} \in \mathbb{C}^{N_r \times N_u}$ is the channel matrix, $\bar{\mathbf{z}} \sim \mathcal{CN}(\mathbf{0}, \sigma_0^2 \mathbf{I}_{N_r})$ is a vector of complex circular Gaussian noise, $\bar{\mathbf{x}} \in \mathcal{X}^{N_u}$ is the vector of transmitted symbols, $\mathcal{X}$ is a finite set of constellation points, and $\mathbf{y} \in \mathbb{C}^{N_r}$ is the received vector. Throughout this work we consider an equivalent real-valued representation obtained by considering the real $\Re(.)$ and imaginary $\Im(.)$ parts separately. Denote $\mathbf{x} = [\Re(\bar{\mathbf{x}})^\top \Im(\bar{\mathbf{x}})^\top]$, $\mathbf{y} = [\Re(\bar{\mathbf{y}})^\top \Im(\bar{\mathbf{y}})^\top]$, $\mathbf{z} = [\Re(\bar{\mathbf{z}})^\top \Im(\bar{\mathbf{z}})^\top]$ and

$$\mathbf{H} = \begin{bmatrix} \Re(\bar{\mathbf{H}}) & -\Im(\bar{\mathbf{H}}) \\ \Im(\bar{\mathbf{H}}) & \Re(\bar{\mathbf{H}}) \end{bmatrix}. \tag{3.2}$$

The system can be rewritten in the equivalent real-valued representation as follows*

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{z}, \tag{3.3}$$

In this work, a quadrature amplitude modulation (QAM) is used and symbols are normalized to attain unit average power. Every symbol is assumed to have the same probability of being chosen by the users. Also, although we adopt a common constellation $\mathcal{X}$ for all users in our baseline problem formulation, we relax this requirement in our numerical experiments (see Section 3.5). Moreover, perfect channel state information (CSI) is assumed, which means that $\mathbf{H}$ and σ_0^2 are known at the receiver, with the exception of Section 3.5.7.† Under this setting, the MIMO detection problem can be defined as follows.

Problem 1 Given perfect CSI and an observed $\mathbf{y}$ following (3.3), find an estimate of $\mathbf{x}$.

---

*We define the noise as complex Gaussian circularly symmetric noise with variance σ_0^2, so each component has variance $\sigma_0^2/2$. However, we will consider throughout the work that σ_0 is the variance of each component as it is just a matter of re-scaling.

†To avoid notation overload, we adopt the convention that whenever we assume $\mathbf{H}$ to be known, σ_0^2 is also known.

Given the stochastic nature of $\mathbf{z}$ in (3.3), a natural way of solving Problem 1 is to search for the $\mathbf{x}$ that maximizes its *posterior* probability given the observations $\mathbf{y}$. Hence, the optimal decision rule can be written as

$$\begin{aligned}\hat{\mathbf{x}}_{\text{MAP}} &= \underset{\mathbf{x}\in\mathcal{X}^{N_u}}{\text{argmax}}\ p(\mathbf{x}|\mathbf{y},\mathbf{H}) \\ &= \underset{\mathbf{x}\in\mathcal{X}^{N_u}}{\text{argmax}}\ p_{\mathbf{z}}(\mathbf{y}-\mathbf{H}\mathbf{x})p(\mathbf{x}),\end{aligned} \tag{3.4}$$

where we applied the Bayes' rule. We assume that the symbols' prior distribution is uniform among the constellation elements and the measurement noise $\mathbf{z}$ is Gaussian. As a result, the maximum a posteriori (MAP) detector reduces to an ML detector. Specifically, (3.4) reduces to solving the following optimization problem

$$\hat{\mathbf{x}}_{\text{ML}} = \underset{\mathbf{x}\in\mathcal{X}^{N_u}}{\text{argmin}}\ ||\mathbf{y}-\mathbf{H}\mathbf{x}||_2^2, \tag{3.5}$$

which is NP-hard due to the finite constellation constraint $\mathbf{x}\in\mathcal{X}^{N_u}$, rendering $\hat{\mathbf{x}}_{\text{ML}}$ intractable for practical applications. Consequently, several schemes have been proposed in the past decades to provide efficient approximate solutions to Problem 1. We detailed some of them in Section 3.2.

3.2 Related work

Many MIMO detection algorithms that approximate the optimal solution (3.5) are iterative, where each iteration consists of the following two general steps

$$\mathbf{u}_t = \hat{\mathbf{x}}_t + \mathbf{A}_t(\mathbf{y}-\mathbf{H}\hat{\mathbf{x}}_t) + \mathbf{b}_t, \tag{3.6a}$$

$$\hat{\mathbf{x}}_{t+1} = \eta_t(\mathbf{u}_t, \sigma_0^2). \tag{3.6b}$$

The first *linear* step (3.6a) takes as input the tuple $(\mathbf{y},\mathbf{H},\hat{\mathbf{x}}_t)$ and applies a linear transformation $\mathbf{A}_t$ to obtain an intermediate signal $\mathbf{u}_t$ in a continuous domain. The

second *non-linear* step (3.6b) applies a *denoiser* η_t to the intermediate signal to get the estimation $\hat{\mathbf{x}}_{t+1}$ of the next iteration. Hence, the estimation is composed by an optimization in a continuous domain followed by a (possibly approximate) projection to the discrete constellation set.

A common choice for the denoiser is the minimizer of $\mathbb{E}[||\eta_t(\mathbf{u}_t, \sigma_0^2) - \mathbf{x}||_2 | \mathbf{u}_t]$, given by the conditional expectation $\eta_t(\mathbf{u}_t, \sigma_0^2) = \mathbb{E}[\mathbf{x}|\mathbf{u}_t]$. For the i.i.d. Gaussian noise case, it boils down to an element-wise function with the following expression

$$\mathbb{E}[x_j|[\mathbf{u}_t]_j] = \frac{1}{Z} \sum_{x_k \in \mathcal{X}} x_k \exp\left(\frac{-||[\mathbf{u}_t]_j - x_k||^2}{2\sigma_0^2}\right), \tag{3.7}$$

where $Z = \sum_{x_k \in \mathcal{X}} \exp\left(\frac{-||[\mathbf{u}_t]_j - x_k||^2}{2\sigma_0^2}\right)$. Notice that (3.7) depends on $\mathbf{u}_t$ and the *noise variance* σ_0. We now describe several detectors, both classical and learning-based, and we show that many of these approximate solutions represent particular cases of (3.6).

3.2.1 Classical MIMO detectors.

The classical body of work (ZF, MMSE, AMP) can be described as particular cases of the framework in (3.6).

Zero-forcing and MMSE detector. ZF is equivalent to a single-step in (3.6a), where $\mathbf{A}_0 = \mathbf{H}^\dagger, \mathbf{b}_0 = 0$ and $\hat{\mathbf{x}}_0 = 0$. MMSE detector, which corresponds to regularized version of ZF, is described by setting $\mathbf{A}_0 = (\mathbf{H}^\top \mathbf{H} + \sigma_0^2 \mathbf{I}_{2N_u})^{-1}, \mathbf{b}_0 = 0$ and $\hat{\mathbf{x}}_0 = 0$. Both methods solve first a relaxation of the original problem, by defining $\mathbf{x} \in \mathcal{X}^{N_u}$ as $\mathbf{x} \in \mathbb{C}^{N_u}$. After this, the final solution is obtained by rounding the intermediate estimation onto one of the constellation elements. Although both are practical because of their low complexity, their performance degrades in massive MIMO systems.

AMP. A more sophisticated classical method is AMP, which can also be described as a particular case of (3.6), where $\mathbf{A}_t = \mathbf{H}^H$ and $\mathbf{b}_t$ is known as the Onsager term [34]. This method represents the MIMO system as a graphical model and performs approximate inference instead of belief propagation as it is prohibitive for massive MIMO systems. It presents a good balance between complexity and performance, as it is asymptotically optimal with i.i.d. Gaussian channels and it has low complexity. However, its behavior depends on the channel model, so its performance degrades significantly for other channel matrices that are not i.i.d. Gaussian. As a consequence, a generalization of AMP, known as Orthogonal AMP (OAMP), was proposed in [35]. It relaxes the constraint on channel matrices imposed by AMP to unitary-invariant matrices [36].

Other detectors. Other low-complexity detectors are V-BLAST [37] and Semi-Definite Relaxation (SDR) [38]. However, neither of the methods balances low complexity and good performance to be on par with the optimal detector (ML).

While classical methods seek to solve the optimization problem, an alternative solution to Problem 1 is to generate samples of the posterior distribution. This can be done using MCMC methods [7, Chapter 8]. For instance, a detector based on the Gibbs Sampler was presented in [39]. Recently, in [40], a detector based on Metropolis-Hastings was proposed. The idea of this detector is to explore the search space following a random walk in the gradient descent direction. Notice that this method can be described by (3.6) by adding a randomized step to generate random samples. Lastly, a detector using the Langevin dynamics was independently proposed in [41] (concurrently made available with our preliminary work [42]), but the approach is significantly different from the one presented in this paper, as they do not adopt

an annealing process to incorporate the discrete nature of the problem, they are constrained to a particular modulation scheme, and they do not propose a robust learning-based version rooted in algorithm unfolding.

3.2.2 Learning-based MIMO detectors.

The application of deep learning to MIMO detection is a body of work that has gained traction in the last few years. The basic idea is to learn a map – a function approximator – from the space of observations $\mathbf{y}$ and CSI $\mathbf{H}$ to the corresponding (approximate) transmitted symbols $\mathbf{x}$. In this way, when a new observation $\mathbf{y}$ is received (along with the CSI), $\mathbf{x}$ can be efficiently estimated using the learned map. Based on how the learned map is designed, one can roughly categorize each detector as either data-driven or model-driven detector.

MMNet. MMNet [43] is a low complexity channel-specific detector based on the iterative soft-thresholding algorithm [44]. It can be described as in (3.6) by replacing $\mathbf{A}_t$ with a matrix of learning parameters, setting $\mathbf{b}_t = 0$, and adding a vector of learning parameters to the noise estimation in the denoiser (3.7). Although its performance is comparable to the ML for i.i.d. Gaussian channels, for correlated ones the detector needs to be trained online for each channel realization. Hence, it is unsuitable for real-time applications. However, this issue can be alleviated by using HyperNetworks [11, 12].

DetNet. DetNet, proposed in [8], is a data-driven detector inspired by iterative projected gradient descent. Its performance is very good on i.i.d. Gaussian channels with low-order modulation schemes, but it degrades for correlated channels and higher-order modulations.

OAMPNet. OAMPNet [10] is a model-driven detector based on the OAMP algorithm. It adds two learning parameters per iteration of the OAMP algorithm. For i.i.d. Gaussian channels, its performance is similar to the ML. However, it performs poorly in more realistic channels. Similar to MMNet, it can be described by (3.6).

RE-MIMO. RE-MIMO [9] is a detector inspired by the Recurrence Inference Machine framework [45] that achieves state-of-the-art performance for correlated channel models. It is an iterative detector with two steps. Each iteration is composed by an encoder step, parameterized by a transformer, and a predictor module, which plays the role of the denoiser. In addition to its good performance, the detector can handle a varying number of transmitters using the same trained model. However, the number of parameters is high, and from the practical point of view, it does not handle different modulations simultaneously, and its performance degrades severally under noise uncertainty.

3.3 Detection by sampling from the posterior distribution

Problem 1 is intractable due to the finite constellation constraint. Therefore, instead of solving the problem using an approximate solution like the ones described in Section 3.2, we propose to *generate a set of samples that approximately come from the posterior distribution* $p(\mathbf{x}|\mathbf{y},\mathbf{H})$ *using* (2.6) *and then select the one that minimizes the objective in* (3.5). The key ingredient in the Langevin dynamic is the score function, which for our case is given by $\nabla_{\mathbf{x}} \log p(\mathbf{x}|\mathbf{y},\mathbf{H})$. After applying Bayes' rule, this score function can be rewritten as

$$\nabla_{\mathbf{x}} \log p(\mathbf{x}|\mathbf{y},\mathbf{H}) = \nabla_{\mathbf{x}} \log p(\mathbf{y}|\mathbf{x},\mathbf{H}) + \nabla_{\mathbf{x}} \log p(\mathbf{x}), \tag{3.8}$$

where the term $\nabla_{\mathbf{x}} \log p(\mathbf{y}|\mathbf{x}, \mathbf{H})$ corresponds to the score function of the likelihood and $\nabla_{\mathbf{x}} \log p(\mathbf{x})$ to the score function of the prior. Notice that this latter term is not well defined due to the discrete nature of the symbols.

To circumvent this obstacle, inspired by [46], we approximate the prior by using an *annealed* version of the Langevin dynamic. Specifically, instead of working with the discrete symbols $\mathbf{x}$, we define a perturbed version of the symbols $\tilde{\mathbf{x}} = \mathbf{x} + \mathbf{n}$ with $\mathbf{n} \sim \mathcal{N}(0, \sigma^2 \mathbf{I})$, for different values of σ^2. Note that $\tilde{\mathbf{x}}$ is now continuous, so it is a random variable described by a probability density with a well-defined gradient. This allows us to run a Langevin dynamic in $\tilde{\mathbf{x}}$ instead of $\mathbf{x}$. Moreover, if we make $\sigma^2 \to 0$ then $\tilde{\mathbf{x}}$ concentrates around $\mathbf{x}$, allowing us to effectively sample from $p(\mathbf{x}|\mathbf{y}, \mathbf{H})$, as wanted.

In a nutshell, the algorithm works as follows. First, we initialize $\tilde{\mathbf{x}}_0$ uniformly at random in $[-1, 1] \times [-1, 1]$, since the symbols are assumed to be normalized. Then, we follow the direction of the score function of the log-posterior density of the perturbed symbol $\nabla_{\tilde{\mathbf{x}}} \log p(\tilde{\mathbf{x}}|\mathbf{y}, \mathbf{H})$, starting with a high σ and gradually decreasing its value until $\tilde{\mathbf{x}} \approx \mathbf{x}$. Apart from enabling the approximation of the score function of the prior distribution, the annealing process also improves the mixing time of the Langevin dynamic [24]. This is particularly important in multimodal distributions. Having introduced the high-level idea of our method, we now provide more details on the *annealing process*, exact expressions for the terms in the *score function*, and a step-by-step description of the *algorithm*.

3.3.1 Annealing process.

We define a sequence of noise levels $\{\sigma_l\}_{l=1}^{L+1}$ such that $\sigma_1 > \sigma_2 > \cdots > \sigma_L > \sigma_{L+1} = 0$. Then, at each level we define a perturbed version of the true symbols $\mathbf{x}$

$$\tilde{\mathbf{x}}_l = \mathbf{x} + \mathbf{n}_l, \tag{3.9}$$

where $\mathbf{n}_l \sim \mathcal{N}(0, \sigma_l^2 \mathbf{I})$. A representation (for a QPSK modulation) of this process is shown in Fig. 1.1. From the figure, we observe that the perturbed symbols defined in (3.9) are defined in a continuous domain due the noise addition, and centered at each symbol. Since the variance of the noise injected at each level is a predefined parameter, we design the sequence in such a way that the noise injected in the first levels is high enough to explore the search space, while at the last levels is very small, approximating the true discrete distribution given by a set of delta functions at each symbol with uniform weight. Notice that we defined the last variance σ_{L+1} as equal to 0, although in practice it is never reached. However, it is necessary for the construction of the synthetic noise, which is explained next.

Given the perturbed symbols in (3.9), the forward model in (3.3) can be rewritten as

$$\mathbf{y} = \mathbf{H}\tilde{\mathbf{x}}_l + (\mathbf{z} - \mathbf{H}\mathbf{n}_l). \tag{3.10}$$

In the new forward model in (3.10), the likelihood is given by $p(\mathbf{y}|\tilde{\mathbf{x}}_l, \mathbf{H}) = p(\mathbf{z} - \mathbf{H}\mathbf{n}_l|\tilde{\mathbf{x}}_l)$, which is not Gaussian: although $p(\mathbf{n}_l)$ is a Gaussian distribution, after conditioning on $\tilde{\mathbf{x}}_l$ the conditional distribution $p(\mathbf{n}_l|\tilde{\mathbf{x}}_l)$ is no longer Gaussian. To understand this, notice that (3.9), $p(\mathbf{n}_l|\tilde{\mathbf{x}}_l) \propto p(\tilde{\mathbf{x}}_l|\mathbf{n}_l)p(\mathbf{n}_l)$ is a mixture of Gaussians weighted by the distribution of $\mathbf{x}$.

However, an analytical expression for the score of the likelihood can still be obtained by following the approach in [46], where a synthetic annealed noise, that is

carved from the measurement noise $\mathbf{z}$, is constructed gradually. Hence, before giving an expression of the score functions, we explain how the synthetic noise is defined. Before moving to this explanation, and in order to get a tractable expression, we have to rely on the singular value decomposition (SVD) of the channel matrix given by $\mathbf{H} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top$ as well as in the spectral representation of $\tilde{\mathbf{x}}_l$, $\mathbf{y}$, $\mathbf{z}$ and $\mathbf{n}_l$ defined as $\tilde{\boldsymbol{\chi}}_l = \mathbf{V}^\top\tilde{\mathbf{x}}_l$, $\boldsymbol{\eta} = \mathbf{U}^\top\mathbf{y}$, $\boldsymbol{\zeta} = \mathbf{U}^\top\mathbf{z}$ and $\boldsymbol{\nu}_l = \mathbf{V}^\top\mathbf{n}_l$. Given the spectral representation, the model in (3.10) can be rewritten as

$$\boldsymbol{\eta} = \boldsymbol{\Sigma}\tilde{\boldsymbol{\chi}}_l + (\boldsymbol{\zeta} - \boldsymbol{\Sigma}\boldsymbol{\nu}_l). \tag{3.11}$$

The model defined in (3.11) is equivalent to (3.10) but in the SVD (spectral) domain. Notice that expressing the system in the spectral domain as in (3.11) maintains the distributions involved invariant because $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices.

3.3.2 Synthetic noise construction.

The key idea of the synthetic noise is that it is carved from $\mathbf{z}$ gradually. To see what we formally mean by this, we assume that for every j such that $s_j \neq 0$, with s_j a singular value of $\mathbf{H}$, there exists an l_j such that $\sigma_{l_j} s_j < \sigma_0$ and $\sigma_{l_j-1} s_j > \sigma_0$. Then, we define $\tilde{\mathbf{x}}_{L+1} = \mathbf{x}$ and for every $l = L, L-1, \cdots, 1$, $\tilde{\mathbf{x}}_l = \tilde{\mathbf{x}}_{l+1} + \boldsymbol{\omega}_l$ where $\boldsymbol{\omega}_l \sim \mathcal{N}(0, (\sigma_l^2 - \sigma_{l+1}^2)\mathbf{I})$. Notice that with this definition, we recover the definition in (3.9) by replacing recursively $\tilde{\mathbf{x}}_{l+1}$ and where $\mathbf{n}_l = \sum_{k=l}^{L} \boldsymbol{\omega}_k$. Thus, the noise in (3.11) can be rewritten as

$$\boldsymbol{\zeta} - \boldsymbol{\Sigma}\boldsymbol{\nu}_l = \boldsymbol{\zeta} - \boldsymbol{\Sigma}\Big(\sum_{k=l}^{L} \mathbf{V}^\top\boldsymbol{\omega}_k\Big), \tag{3.12}$$

Given (3.12), it follows that the annealed noise $\boldsymbol{\nu}_l$ is independent of $\tilde{\mathbf{x}}_l$; for more details, see [46, Appendix A]. Moreover, given that both $\boldsymbol{\zeta}$ and $\mathbf{V}^\top\boldsymbol{\omega}_k$ are Gaussian,

the expression in (3.12) is also Gaussian with 0 mean and a covariance matrix given by

$$\mathbb{E}[(\boldsymbol{\zeta} - \boldsymbol{\Sigma}\boldsymbol{\nu}_l)(\boldsymbol{\zeta} - \boldsymbol{\Sigma}\boldsymbol{\nu}_l)^\top] = \mathbb{E}[\boldsymbol{\zeta}\boldsymbol{\zeta}^\top] + \boldsymbol{\Sigma}\mathbb{E}[\boldsymbol{\nu}_l\boldsymbol{\nu}_l^\top]\boldsymbol{\Sigma}^\top - 2\mathbb{E}[\boldsymbol{\Sigma}\boldsymbol{\nu}_l\boldsymbol{\zeta}^\top. \tag{3.13}$$

The first two terms are known – they are the covariance matrices of the measurement noise and the noise defined in (3.9) –, and the third term can be rewritten as $\mathbb{E}[\boldsymbol{\Sigma}\boldsymbol{\nu}_l\boldsymbol{\zeta}^\top] = \mathbb{E}[\boldsymbol{\Sigma}(\sum_{k=l}^{L}\mathbf{V}^\top\boldsymbol{\omega}_k)\boldsymbol{\zeta}^\top] = \sum_{k=l}^{L}\mathbb{E}[\boldsymbol{\Sigma}\mathbf{V}^\top\boldsymbol{\omega}_k\boldsymbol{\zeta}^\top]$.

Hence, we need to define the statistical relationship between $\boldsymbol{\zeta}$ and $\boldsymbol{\Sigma}\mathbf{V}^\top\boldsymbol{\omega}_k$, which is the key step in this construction. Essentially, we seek for an uncorrelated multivariate Gaussian distribution where the variance depends only on the relation between $\sigma_l s_j$ and σ_0, i.e., each direction in the spectral domain is independent of each other. Hence, following [46], we define the noise as (we focus on a single entry j)

$$[\mathbb{E}[\boldsymbol{\Sigma}\mathbf{V}^\top\boldsymbol{\omega}_l\boldsymbol{\zeta}^\top]]_j = \begin{cases} [\mathbb{E}[(\boldsymbol{\Sigma}\mathbf{V}^\top\boldsymbol{\omega}_l)(\boldsymbol{\Sigma}\mathbf{V}^\top\boldsymbol{\omega}_l)^\top]]_j, & l \geq l_j \\ [\mathbb{E}[(\boldsymbol{\zeta} - \boldsymbol{\Sigma}\boldsymbol{\nu}_{l_j})(\boldsymbol{\zeta} - \boldsymbol{\Sigma}\boldsymbol{\nu}_{l_j})^\top]]_j, & l = l_j - 1 \\ 0, & l < l_j - 1. \end{cases} \tag{3.14}$$

It can be checked by replacing (3.14) in (3.13) that the distribution of $\boldsymbol{\zeta} - \boldsymbol{\Sigma}\boldsymbol{\nu}_l$ is multivariate Gaussian, where each component is distributed as $[\boldsymbol{\zeta} - \boldsymbol{\Sigma}\boldsymbol{\nu}_l]_j \sim \mathcal{N}(0, |\sigma_0^2 - \sigma_l^2 s_j^2|)$ for $j = 1, \cdots, N_u$. Intuitively, when $\sigma_i s_j < \sigma_0$, the synthetic noise is fully immersed in the measurement noise, while if $\sigma_i s_j > \sigma_0$ there is an independent portion that is added.

3.3.3 Score functions.

With the spectral representation in mind, our goal is to run a Langevin dynamic whose score function for every noise level l is given by [cf. (3.8)]

$$\nabla_{\tilde{\boldsymbol{\chi}}_l}\log p(\tilde{\boldsymbol{\chi}}_l|\boldsymbol{\eta}, \mathbf{H}) = \nabla_{\tilde{\boldsymbol{\chi}}_l}\log p(\boldsymbol{\eta}|\tilde{\boldsymbol{\chi}}_l, \mathbf{H}) + \nabla_{\tilde{\boldsymbol{\chi}}_l}\log p(\tilde{\boldsymbol{\chi}}_l). \tag{3.15}$$

We now provide a closed-form expression for both constituent terms in this score function.

i) Score of the likelihood: Based on the above discussion, the final expression for the score of the likelihood in the spectral domain is given by

$$\nabla_{\tilde{\boldsymbol{\chi}}_l} \log p(\boldsymbol{\eta}|\tilde{\boldsymbol{\chi}}_l, \mathbf{H}) = \boldsymbol{\Sigma}^\top \ |\sigma_0^2 \mathbf{I} - \sigma_l^2 \boldsymbol{\Sigma}\boldsymbol{\Sigma}^\top|^\dagger \ (\boldsymbol{\eta} - \boldsymbol{\Sigma}\tilde{\boldsymbol{\chi}}_l). \tag{3.16}$$

To give some intuition, the score function of the likelihood is given by the gradient of a multivariate Gaussian distribution: the residual error $(\boldsymbol{\eta} - \boldsymbol{\Sigma}\tilde{\boldsymbol{\chi}}_l) = (\mathbf{U}^\top \mathbf{y} - \boldsymbol{\Sigma}\mathbf{V}^\top \tilde{\mathbf{x}}_l)$ is multiplied by the (pseudo-)inverse of the covariance matrix, which is a diagonal matrix with entries given by $|\sigma_0^2 - \sigma_l^2 s_j^2|$.

ii) Score of the annealed prior: We first notice that $\nabla_{\tilde{\boldsymbol{\chi}}_l} \log p(\tilde{\boldsymbol{\chi}}_l) = \mathbf{V}^\top \nabla_{\tilde{\mathbf{x}}_l} \log p(\tilde{\mathbf{x}}_l)$ due to the orthogonality of $\mathbf{V}$. Moreover, based on the Tweedie's identity [47], we can relate the score function $\nabla_{\tilde{\mathbf{x}}_l} \log p(\tilde{\mathbf{x}}_l)$ and the MMSE denoiser as follows

$$\nabla_{\tilde{\mathbf{x}}_l} \log p(\tilde{\mathbf{x}}_l) = \frac{\mathbb{E}_{\sigma_l}[\mathbf{x}|\tilde{\mathbf{x}}_l] - \tilde{\mathbf{x}}_l}{\sigma_l^2}. \tag{3.17}$$

In particular, the conditional expectation, defined as $\mathbb{E}_{\sigma_l}[\mathbf{x}|\tilde{\mathbf{x}}_l] = \sum_{\mathbf{x}_k \in \mathcal{X}^{N_u}} \mathbf{x}_k p(\mathbf{x}_k|\tilde{\mathbf{x}}_l)$ $= \frac{1}{p(\tilde{\mathbf{x}}_l)} \sum_{\mathbf{x}_k \in \mathcal{X}^{N_u}} \mathbf{x}_k p(\tilde{\mathbf{x}}_l|\mathbf{x}_k) p(\mathbf{x}_k)$, can be calculated element-wise as

$$\mathbb{E}_{\sigma_l}[x_j|[\tilde{\mathbf{x}}_l]_j] = \frac{1}{Z} \sum_{x_k \in \mathcal{X}} x_k \exp\left(\frac{-||[\tilde{\mathbf{x}}_l]_j - x_k||^2}{2\sigma_l^2}\right), \tag{3.18}$$

where $Z = \sum_{x_k \in \mathcal{X}} \exp\left(\frac{-||[\tilde{\mathbf{x}}_l]_j - x_k||^2}{2\sigma_l^2}\right)$ and $j = 1, \cdots, N_u$.

3.3.4 Langevin-based MIMO detector algorithm.

Alg. 1 illustrates how to generate samples $\hat{\mathbf{x}}$ from the (approximate) posterior $p(\mathbf{x}|\mathbf{y}, \mathbf{H})$. Notice that we write the Langevin equation with the temperature parameter τ, as it is more general; when $\tau = 1$, the dynamic boils down to the annealed Langevin dynamic in (2.6). As discussed in [46], when computing the entries of the score function

in (3.15) using the expressions in (3.16) and (3.17), one of these terms might be negligible with respect to the other depending on the noise level. Thus, the elementwise score of the posterior will be given by

$$[\nabla_{\tilde{\boldsymbol{\chi}}_l} \log p(\tilde{\boldsymbol{\chi}}_l|\boldsymbol{\eta}, \mathbf{H})]_j = \tag{3.19}$$

$$\begin{cases} [\nabla_{\tilde{\boldsymbol{\chi}}_l} \log p(\boldsymbol{\eta}|\tilde{\boldsymbol{\chi}}_l, \mathbf{H}) + \mathbf{V}^\top \nabla_{\tilde{\mathbf{x}}} \log p(\tilde{\mathbf{x}})]_j, & \sigma_0 \geq \sigma_l s_j \\ [\nabla_{\tilde{\boldsymbol{\chi}}_l} \log p(\boldsymbol{\eta}|\tilde{\boldsymbol{\chi}}_l, \mathbf{H})]_j, & \sigma_0 < \sigma_l s_j \\ [\mathbf{V}^\top \nabla_{\tilde{\mathbf{x}}} \log p(\tilde{\mathbf{x}})]_j, & s_j = 0. \end{cases}$$

Intuitively, when the injected noise at level l is such that $\sigma_l s_j > \sigma_0$, then the contribution of the score of the prior is negligible and can be ignored, meaning that the Langevin dynamic considers *the observation as a denoised estimation*. Then, when $\sigma_l s_j < \sigma_0$, the estimation needs to be refined by *considering prior information combined with the likelihood term*. Similarly, whenever $s_j = 0$, the corresponding entry η_j is uninformative and the score of the likelihood can be ignored. Furthermore, a refinement that we incorporate in the algorithm is the use of user-dependent step sizes. Instead of using a constant *scalar* step size ϵ as in (2.6) or even time-varying versions of it, in Alg. 1 we employ user-dependent *diagonal matrices* $\boldsymbol{\Lambda}_l$. In this way, different entries of our vector-valued Langevin dynamic can be updated at different rates depending on the singular values of the channel under consideration. Finally, given that $\sigma_L \neq 0$, the sample will be very close to the constellation but not exactly. Hence, we project our sample to the constellation by taking $\bar{\mathbf{x}} = \text{argmin}_{\mathbf{x} \in \mathcal{X}^{N_u}} ||\mathbf{x} - \mathbf{V}\tilde{\boldsymbol{\chi}}_{T,L}||_2^2$.

Given that Alg. 1 is stochastic, one can run the algorithm multiple times to generate several samples $\bar{\mathbf{x}}$ from the same (approximate) posterior distribution. Hence, as we want to approximate the MAP estimate – equivalently for this case, the ML estimate – we run M different Langevin trajectories for each pair $\{\mathbf{y}, \mathbf{H}\}$ and keep

the sample that minimizes (3.5). Formally, given M samples $\{\bar{\mathbf{x}}_m\}_{m=1}^M$ obtained from Alg. 1, our final estimate is given by

$$\hat{\mathbf{x}} = \underset{\mathbf{x}\in\{\bar{\mathbf{x}}_m\}_{m=1}^M}{\operatorname{argmin}} ||\mathbf{y} - \mathbf{H}\mathbf{x}||_2^2. \tag{3.20}$$

Notice that these M trajectories can be run in parallel, as they are independent of each other.

Computational complexity. The first step in Alg. 1 is to compute the SVD of the channel $\mathbf{H}$, whose complexity is $\mathcal{O}(N_u N_r \min\{N_u, N_r\})$, and is done only once per channel. Moreover, notice that the matrices involved in each iteration are diagonal so the complexity of multiplying them is $\mathcal{O}(N_u^2)$, while the matrix inversion is $\mathcal{O}(N_u)$. Finally, given a modulation of K symbols, the complexity of (4.10) is $\mathcal{O}(KN_u)$. Hence, one iteration has a complexity of $\mathcal{O}(N_u^2 + KN_u)$. The overall complexity, including the SVD computation and all the iterations, is $\mathcal{O}(N_u N_r \min\{N_u, N_r\} + LT(N_u^2 + KN_u))$. Regarding the M trajectories, observe that these are independent of each other, so they can be computed in parallel. To given some context about the performance, we show in Table 3.1 the complexity of the baseline detectors that are used in the numerical experiments[‡]. We observe that our method has also a polynomial complexity, and has a similar complexity as long as the number of noise levels L and iterations per level T are controlled and pushed to small values. Therefore, the bottlenecks are twofold: the SVD computation and the number of iterations LT. While the former is inevitable, the latter is a parameter that we control and represents a trade-off between SER performance and computational complexity. In Section 3.5,

[‡]For RE-MIMO, each layer corresponds a block of an encoder module given by a Transformer with a fixed number of layers, and a predictor module parameterized by an MLP. See [9] for more details.

we present numerical experiments that analyze this trade-off and the impact on the SER performance.

Algorithm 1 Annealed Langevin for MIMO detection

Require: $T, \{\sigma_l\}_{l=1}^{L}, \epsilon, \sigma_0, \mathbf{H}, \mathbf{y}, \tau$

Compute SVD of $\mathbf{H} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top$

Initialize $\tilde{\boldsymbol{\chi}}_{t=0,l=1}$ with random noise $\mathcal{U}[-1, 1]$

for $l = 1$ to L **do**

$$[\boldsymbol{\Lambda}_l]_{jj} = \begin{cases} \frac{\epsilon\sigma_l^2}{\sigma_L^2}(1 - \frac{\sigma_l^2}{\sigma_0^2}s_j^2) & \text{if } \sigma_l s_j \leq \sigma_0 \\ \frac{\epsilon}{\sigma_L^2}(\sigma_l^2 - \frac{\sigma_0^2}{s_j^2}) & \text{if } \sigma_l s_j > \sigma_0 \end{cases}$$

for $t = 0$ to $T - 1$ **do**

Draw $\mathbf{w}_t \sim \mathcal{N}(0, \mathbf{I})$

Compute $\nabla_{\tilde{\boldsymbol{\chi}}_{t,l}} \log p(\boldsymbol{\eta}|\tilde{\boldsymbol{\chi}}_{t,l}, \mathbf{H})$ as in (3.16)

Compute $\nabla_{\tilde{\mathbf{x}}_{t,l}} \log p(\tilde{\mathbf{x}}_{t,l})$ as in (3.17)

Compute $\nabla_{\tilde{\boldsymbol{\chi}}_{t,l}} \log p(\tilde{\boldsymbol{\chi}}_{t,l}|\boldsymbol{\eta}, \mathbf{H})$ as in (3.19)

$\tilde{\boldsymbol{\chi}}_{t+1,l} = \tilde{\boldsymbol{\chi}}_{t,l} + \boldsymbol{\Lambda}_l \nabla_{\tilde{\boldsymbol{\chi}}_{t,l}} \log p(\tilde{\boldsymbol{\chi}}_{t,l}|\boldsymbol{\eta}, \mathbf{H}) + \sqrt{2\boldsymbol{\Lambda}_l\tau}\, \mathbf{w}_t$

end for

$\tilde{\boldsymbol{\chi}}_{0,l+1} = \tilde{\boldsymbol{\chi}}_{T,l}$

end for

return $\bar{\mathbf{x}} = \text{argmin}_{\mathbf{x}\in\mathcal{X}^{N_u}} ||\mathbf{x} - \mathbf{V}\tilde{\boldsymbol{\chi}}_{T,L}||_2^2$

3.4 Unfolding of the Langevin detector

In this section, we present an unfolded MIMO detector inspired on the Langevin detector presented in Section 2.1. We briefly introduce the algorithm unfolding framework in Section 3.4.1 while in Section 3.4.2 we explain our proposed architecture.

3.4.1 Background in algorithm unfolding

Algorithm unfolding (or unrolling) [48–50] refers to the general notion of selecting an iterative model-based approach to solve a problem of interest and building a problem-specific neural network with layers inspired by these iterations. Based on its recent success on many problems, we propose to apply this framework to the annealed Langevin dynamic explained in Section 2.1 by learning the score of the likelihood.

In other contexts, several previous works have already proposed to parameterize the score function by a *score network*. In [24,26], the authors proposed to parameterize the score function of the distribution and train using denoising score matching [23]. In [46], instead of working with a general distribution, they focus on sampling from a posterior distribution. Therefore, they approximate the score of the prior by a neural network, but the training process is done similarly as in [24]. In this work, we leverage the algorithm unfolding framework to train our proposed architecture end-to-end. In particular, given that we have access to a set of pilot signals, we can train in a supervised way by minimizing the error between our estimation and the true symbol.

Method	Complexity
MMSE	$\mathcal{O}(N_u^3 + N_r N_u^2)$
OAMPNet	$\mathcal{O}(N_r^3 + N_u^3 + N_r N_u^2 + N_u N_r^2)$ per layer
RE-MIMO	$\mathcal{O}(N_r^2 N_u + N_r N_u^2)$ per EP block
Langevin	$\mathcal{O}(N_u N_r \min\{N_u, N_r\} + LT(N_u^2 + K N_u))$

Table 3.1 : Complexity of Langevin and the other baseline detectors. We observe that our proposed detector has a polynomial complexity much like the other detectors.

3.4.2 Detection by sampling from the posterior distribution using a learned score network.

The main objective of the proposed learning-based detector is to be robust against noise variance uncertainty while following the same structure as the detector in Section 2.1. In general, learning-based detectors need access to the true noise variance as they use this parameter in the denoising step (3.6b). However, in our proposed detector we avoid this denoising step by following the annealed Langevin dynamic. Therefore, the only noise variance-dependent term in our detector is the score of the likelihood in (3.16), so we propose to parameterize it by a *score network*. We detail now the architecture of our proposed learning-based detector.

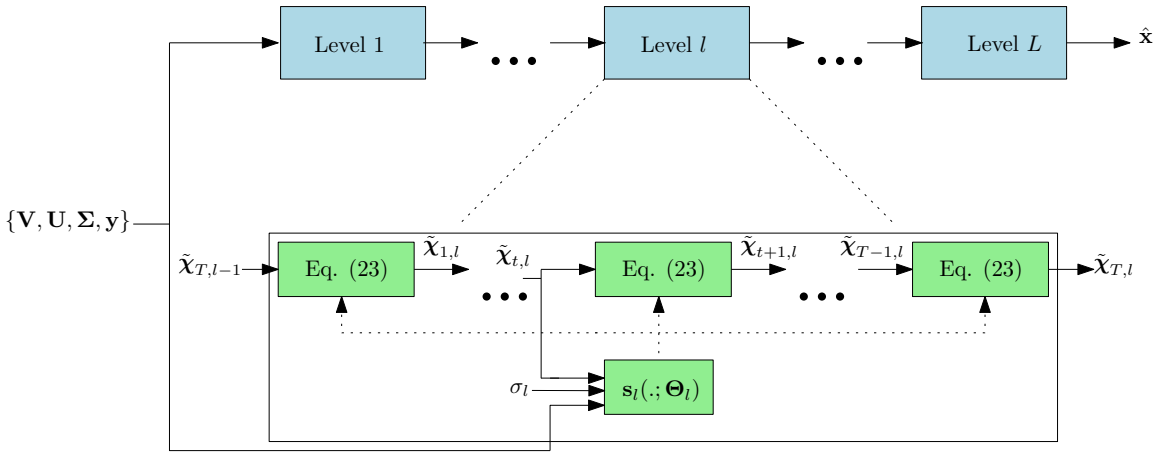


Figure 3.1 : Scheme of our proposed unfolded Langevin detector. The input to architecture is the SVD of $\mathbf{H}$ and the observation $\mathbf{y}$ and the estimated symbol is computed by running sequentially the Langevin dynamic for each of the L levels, represented by blue. Inside each level l, we run (3.21) for T iterations, where the score network $\mathbf{s}_l(.;\mathbf{\Theta})$ is the same for all the interatios in the level. This is represented by green, and the output $\tilde{\boldsymbol{\chi}}_{T,l}$ is the initial value for the next level $l-1$.

Architecture. Our unfolded architecture (depicted in Fig. 3.1) is composed of L noise levels as the detector derived in Section 2.1, where the score of the likelihood is replaced by a *score network*. Formally, we learn a function $\Phi : \mathbb{C}^{N_r} \times \mathbb{C}^{N_r \times N_u} \to \mathcal{X}^{N_u}$ such that $\Phi(\mathbf{H}, \mathbf{y}; \boldsymbol{\Theta})$ is a solution of (3.5). Each layer l of $\Phi(\mathbf{H}, \mathbf{y}; \boldsymbol{\Theta})$, defined as $\phi_l(\mathbf{H}, \mathbf{y}, \tilde{\boldsymbol{\chi}}_{T,l-1}; \boldsymbol{\Theta}_l)$ where $\boldsymbol{\Theta} = \{\boldsymbol{\Theta}_l\}_{l=1}^{L}$, is composed by T iterations and its output is the initial value for the next layer, i.e., $\tilde{\boldsymbol{\chi}}_{T,l}$. Thus, each iteration is defined as

$$\begin{aligned}\tilde{\boldsymbol{\chi}}_{t+1,l} = \tilde{\boldsymbol{\chi}}_{t,l} + \frac{\epsilon\sigma_l^2}{\sigma_L^2}\Big(&\mathbf{s}_l(\boldsymbol{\eta} - \boldsymbol{\Sigma}\tilde{\boldsymbol{\chi}}_{t,l}, \tilde{\boldsymbol{\chi}}_{t,l}, \sigma_l; \boldsymbol{\Theta}_l) \\ &+ \mathbf{V}^\top \nabla_{\tilde{\mathbf{x}}_{t,l}} \log p(\tilde{\mathbf{x}}_{t,l})\Big) + \sqrt{2\frac{\epsilon\sigma_l^2}{\sigma_L^2}\tau}\, \mathbf{w}_t.\end{aligned} \tag{3.21}$$

The neural network $\mathbf{s}_l(.; \boldsymbol{\Theta}_l)$ is the *score network* at level l and $\boldsymbol{\Theta}_l$ is the corresponding set of learning parameters. Given that the score of the likelihood (3.16) depends on the observations $\mathbf{y}$, the channel $\mathbf{H}$, the sample $\tilde{\boldsymbol{\chi}}_{t,l}$ at iteration t and the noise variance of the annealed noise σ_l, we define the tuple $\{\boldsymbol{\eta} - \boldsymbol{\Sigma}\tilde{\boldsymbol{\chi}}_{t,l}, \tilde{\boldsymbol{\chi}}_l, \sigma_l\}$ as the input of the score network. We decided to work in the spectral domain based on the experimental results, which were better than using $\mathbf{H}$ directly.

In essence, our architecture only replaces the noise variance-dependent term, which is the score of the likelihood, by a neural network, and leverage the score of the prior to generate samples from the constellation without a projection step. Each score network $\mathbf{s}_l$ is a multi-layer perceptron (MLP) with 4 layers: the first has 400 hidden neurons, the second one 350 and the last one 100. With the exception of the last layer, all the others use an exponential linear unit (ELU) function as point-wise non-linearity.

Loss function. Our method concatenates L layers, where each layer has its own learning parameters. We use the average ℓ_2 loss over all the layers. More precisely,

the loss function is defined as

$$\mathcal{L}(\boldsymbol{\Theta}) = \mathbb{E}_{\mathbf{x},\mathbf{y},\mathbf{H}}\left[\frac{1}{L}\sum_{l=1}^{L}||\phi_l(\mathbf{H},\mathbf{y},\tilde{\boldsymbol{\chi}}_{T,l-1};\boldsymbol{\Theta}_l)-\mathbf{x}||_{\ell_2}^2\right], \tag{3.22}$$

and we train the neural networks $\mathbf{s}_l(.;\boldsymbol{\Theta}_l)$ (implicit in (3.22) within ϕ_l) by minimizing the empirical loss. In the numerical results we consider the learning architecture with Gaussian noise. However, no assumptions are made with respect to the noise distribution. Hence, the method can be used also in non-Gaussian cases; the only requirement is having an estimation of the likelihood model, which can be obtained as in [51].

3.5 Results

In this section we present the results of our proposed method.[§] We start by analyzing the SER performance of our detector when considering different noise levels L, different number of trajectories M and different temperature parameter τ. Based on this analysis, we fix the optimal hyper-parameters and we compare our method with both classical and learning-based detectors for different channel models – both simulated and real ones – and settings. We also evaluate our method when users are transmitting with different modulation schemes simultaneously. Finally, the robustness of our unfolded method presented in Section 3.4 is compared with other detectors when there is uncertainty in the noise.

Before moving to the experiments, we first discuss the implementation details of the baseline detectors that are used in the experiments. For the learning-based ones, we briefly explain the training process.

[§]Code to replicate the numerical experiments can be found at `https://github.com/nzilberstein/Langevin-MIMO-detector`.

- **MMSE** [6]: Linear detector, which formally solves first a relaxation of (1), by defining $\mathbf{x} \in \mathcal{X}^{N_u}$ as $\mathbf{x} \in \mathbb{C}^{N_u}$, and projection onto one of the constellation elements.

- **V-BLAST** [52]: Multi-stage interference cancellation BLAST algorithm using Zero-Forcing as the detection stage.

- **MMNet-iid** [43]: Learning based detector that is channel-dependent. Hence, we use as a baseline only for the Rayleigh channel model, as in this case it becomes channel-agnostic. We implement the scheme that has only two scalar parameters per layer with 10 layers [43].

- **RE-MIMO** [9]: Recurrent permutation equivariant neural detector based on an encoder-predictor architecture. The encoder is parameterized by a transformer, while the predector with an MLP. We train as proposed in the paper.

- **OAMPNet** [10]: We use 10 layers as it is proposed in the paper. At each layer, a matrix pseudo inverse is required and has 2 learnable parameters.

- **Langevin detector**: Detector proposed in Section 2.1. The following hyper-parameters are fixed for all the experiments unless otherwise mentioned: ϵ is fixed at 3×10^{-5}, the number of samples per noise level at $T = 70$, the number of noise levels at $L = 20$ between $\sigma_1 = 1$ and $\sigma_{20} = 0.01$. The temperature parameter is fixed at $\tau = 1/2$ and the number of trajectories to $M = 20$.

- **U-Langevin detector**: Detector based on an unfolding of the annealed Langevin dynamic as defined in Section 3.4. The training is performed using Adam optimizer [53] for 15,000 iterations with a batch size of 500 matrices per iteration. The learning rate that starts at $l_r = 10^{-3}$ and is scaled by a factor of 10^{-1}

after each 5000 iterations. We consider the same hyperparameter settings as **Langevin detector**, but with $\tau = 1$.

- **ML**: The optimal solver for (3.5) using a highly-optimized mixed integer programming package Gurobi [54].

In all experiments, the signal-to-noise ratio (SNR) is given by

$$\text{SNR} = \frac{\mathbb{E}[||\mathbf{Hx}||^2]}{\mathbb{E}[||\mathbf{z}||^2]}. \tag{3.23}$$

For all the experiments in the following subsections, we consider two different modulation schemes: 16-QAM and 64-QAM. The simulation environment includes a base station with $N_r = 64$ receiver antennas and $N_u = 32$ single-antenna users, except for the real channel, which is further detailed in Section 3.5.3.

3.5.1 Analysis of Langevin method

In this subsection we consider the Kronecker correlated channel model, which is generated following

$$\mathbf{H} = \mathbf{R}_r^{1/2}\mathbf{H}_e\mathbf{R}_u^{1/2}, \tag{3.24}$$

where $\mathbf{H}_e$ is a Rayleigh fading channel matrix and $\mathbf{R}_r$ and $\mathbf{R}_u$ are the spatial correlation matrices at the receiver and transmitters, respectively. These correlation matrices are generated according to the exponential correlation matrix model with a coefficient $\rho = 0.6$; see [55] for details.

Varying the number of noise levels. In the first experiment, given the sequence of noise levels with variance $\{\sigma_l\}_{l=1}^L$, we fix $\sigma_1 = 1$, $\sigma_L = 0.01$, $M = 20$ trajectories and a temperature $\tau = 1/2$ and change the number of noise levels between them. We consider four cases where $L \in \{5, 10, 15, 20\}$; see Fig. 3.2a. First, notice that the

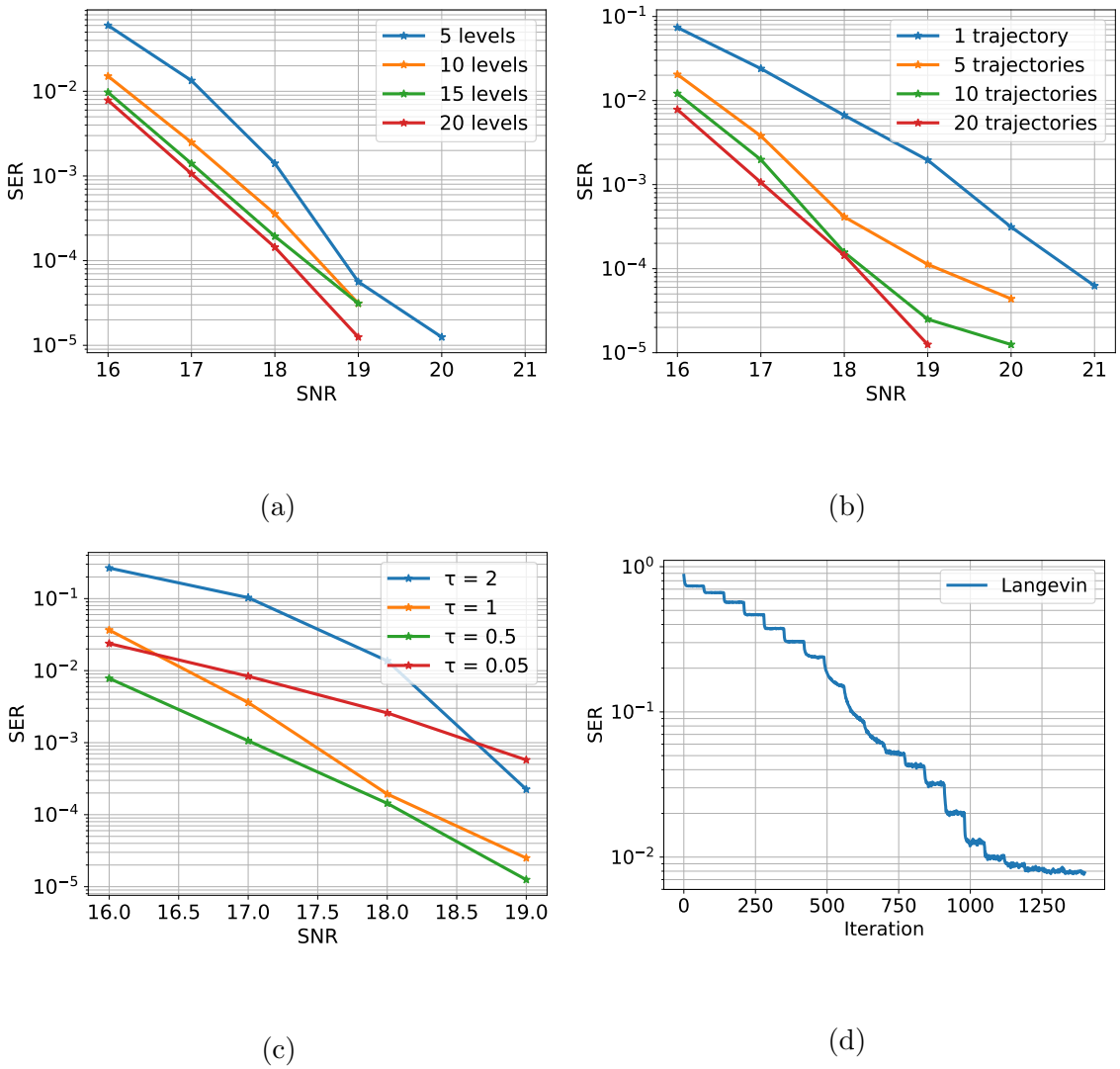


Figure 3.2 : Performance analysis of our proposed method evaluated in a Kronecker correlated channel model as in (3.24). (a) SER as a function of SNR for our Langevin method for noise levels $L \in \{5, 10, 15, 20\}$. (b) SER as a function of SNR for our Langevin method for $M \in \{1, 5, 10, 20\}$ numbers of trajectories. (c) SER as a function of SNR for two different temperature parameters τ. (d) Numerical convergence of our proposed detector, in terms of SER as a function of iterations.

performance, for low SNR when $L = 5$, is much worse than the other three cases. This implies that the algorithm is not able to sufficiently explore the search space. A natural trade-off between computational burden and performance exists: when considering more levels, the SER performance improves at the cost of increasing the running time. From this experiment, we conclude that at least $L = 10$ levels are needed in order to perform as well as the existing state-of-the-art detectors.

Varying the number of trajectories. The performance of the detector as a function of the number M of different Langevin trajectories [cf. (3.20)] is shown in Fig. 3.2b. We analyze five cases where $M \in \{1, 5, 10, 20\}$. For all the cases, we consider $L = 20$ noise levels and $\tau = 1/2$. From the results, we see that M is a hyperparameter that has a high impact on the overall performance of the detector. In particular, if we consider only $M = 1$, then the performance degrades severally, with a SER, for low SNR, in the order of the classical MMSE detector (not shown in the figure). However, if we consider $M = 20$ trajectories, the proposed method outperforms state-of-the-art detectors, as we illustrate in our next experiment.

Varying the temperature parameter. In this experiment, given a fixed sequence of noise levels and $M = 20$ trajectories, we change the temperature parameter τ. We consider four cases where $\tau \in \{2, 1, 0.5, 0.05\}$; see Fig. 3.2c. When $\tau = 1$, we are sampling from the original target distribution as it was described in Section 2.1. While increasing the temperature parameter to 2 increase the mixing time, when the value of τ is decreased to 0.5 the mixing time improves, although we are not sampling from the original posterior distribution. This is related to the landscape of the posterior distribution, as the temperature is scaling the local maxima and concentrating around the global maxima. Lastly, notice that decreasing further the

value of τ has a negative impact, as the mixing time increases.

Convergence result. The last experiment of this subsection is a numerical result for the convergence of our method, where we compute the SER as a function of the iterations. We fixed the parameters to the best combination: a fixed sequence of 20 noise levels, $M = 20$ trajectories, and a temperature parameter $\tau = 0.5$. Moreover, we consider an SNR $= 16\,\text{dB}$. The result is shown in Fig. 3.2d. The total number of iterations is number of noise levels $\times$ number of samples per level, which for our case is $20 \times 70 = 1400$. Therefore, as there 20 levels of noise, there are also 20 jumps between each level in SER. Moreover, the figure reveals that the first and the last levels could be shorter without affecting the performance, as the SER at those levels is almost constant. Thus, the complexity of the detector can be reduced by using an adaptive number of iterations at each level. Although in this work we consider a fixed number of iterations per level for simplicity, we leave the exploration of this direction as future work.

3.5.2 Comparison with other methods

In this subsection we compare our detector with baseline methods. We consider three different channels: Rayleigh MIMO channel, Kronecker correlated MIMO channel and 3GPP channel.

Rayleigh MIMO channel model. In this first case, we consider i.i.d. Gaussian channels. Hence, each element of $\bar{\mathbf{H}}$ is sampled from $\bar{h}_{ij} \sim \mathcal{CN}(0, 1/N_r)$, and then converted to the equivalent real representation.

The comparison for a batch size of 5000 channel realizations is shown in Fig. 3.3a. The figure shows that our method not only outperforms the other detectors, both the

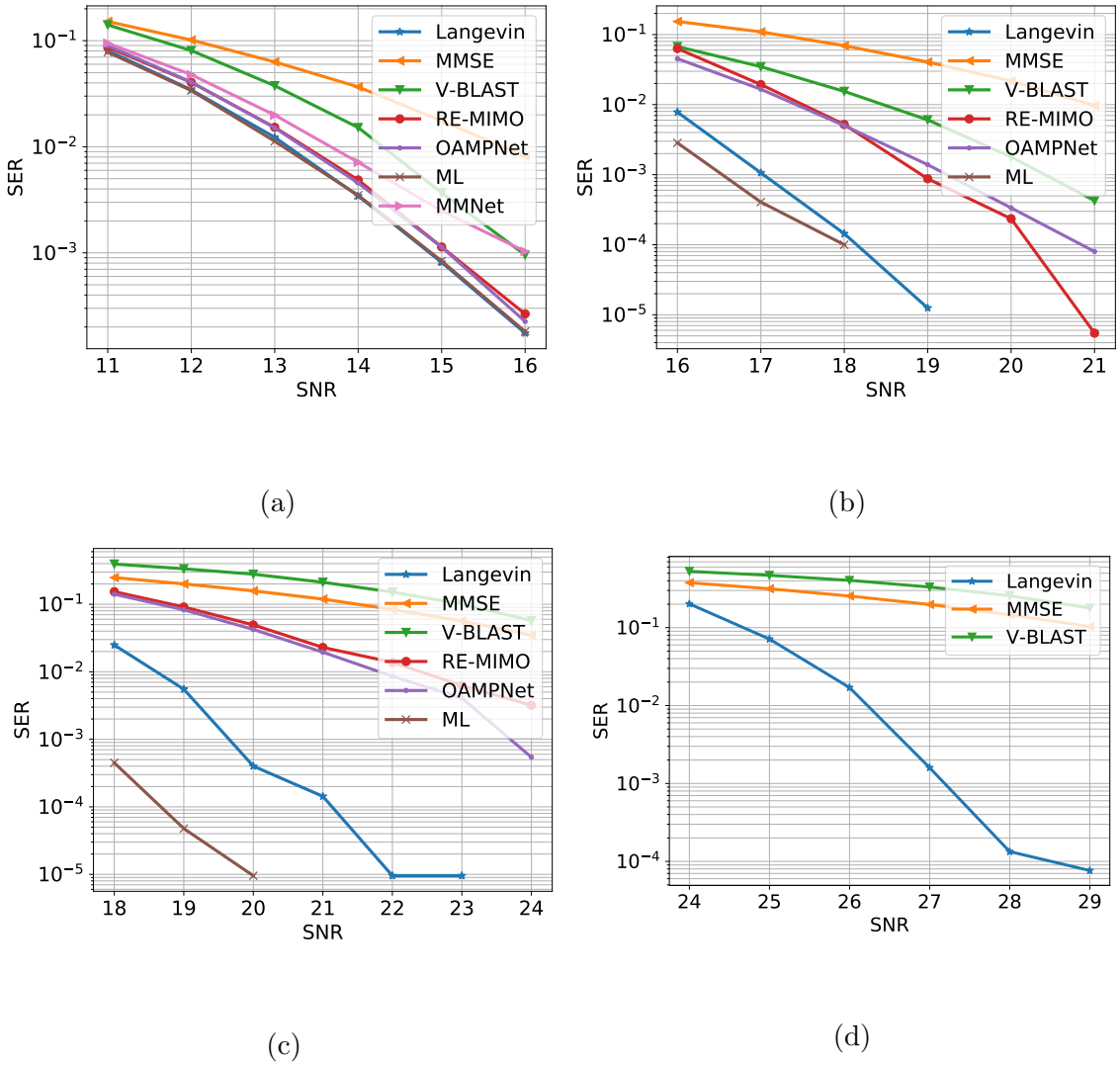


Figure 3.3 : SER as a function of SNR for different detection methods evaluated in (a) Rayleigh channel model and a modulation 16-QAM. (b) Kronecker correlated channel model as in (3.24) and a modulation 16-QAM. (c) 3GPP channel model and a modulation 16-QAM. (d) 3GPP channel model and a modulation 64-QAM.

learning-based and the classical ones, but also achieves the optimal ML performance. In particular, for SNR = 11 dB Langevin achieves an SER of 8.1×10^{-2}, slightly better than its best competitor, RE-MIMO, which achieves an SER of 8.6×10^{-2}, while for

SNR = 16 dB Langevin achieves an SER of 1.7×10^{-4} while the best competitor, OAMPNet, an SER of 2.2×10^{-4}. Furthermore, the gap between Langevin and MMSE increases as the SNR increases. It is particularly interesting to notice that our proposed method outperforms the learning-based detectors, which have been trained with channel instances drawn from the same model as those in the testing set.

Correlated MIMO channel model. In this second case, we consider the channel defined in (3.24). Again, the batch size for testing is 5000 and $\rho = 0.6$.

The comparison is shown in Fig. 3.3b for 16-QAM. The figure reveals that our proposed method markedly outperforms the other detectors. At is explained in [43], in the case of correlated channels, MMNet needs to be trained online for each channel realization correlated channels. This, we do not consider. Again, as in the case of Rayleigh channel, our proposed method outperforms the learning-based detectors, which have been trained with channel instances drawn from the same model as those in the testing set.

3GPP channel model. The last channel model of this subsection is representative of 3GPP 3D MIMO channel model [56], as implemented in the QuaDRiGa channel simulator [57]. We consider a base station with an 8×8 half-wavelength space, single-polarization antenna array at a height of 20 m. We assume that the BS cover a sector of radius 500 m and 32 single-polarization omni-directional antennas users are dropped randomly. Moreover, users are NLOS and indoors. The carrier frequency is 3.5 GHz, and then each subcarrier has a 100 MHz bandwidth. The spacing within subcarriers is 30 kHz. We evaluate the performance in a batch size of 3276 matrices.

The comparison is shown in Figs. 3.3c and 3.3d for a 16-QAM and 64-QAM respectively. For the 64-QAM we only consider the non-learning based method that are

feasible for a real application, as we proved that Langevin outperforms the learning-based. We see that our proposed method still outperforms both classical and learning-based detectors by several orders of magnitude. However, the gap between the ML detector and Langevin detector increases further than for the previous channel models. Notice that we have to increase the SNR ranges by 7 dB and 2 dB with respect to the i.i.d. channel and Kronecker correlated channel respectively, to achieve a similar SER performance.

3.5.3 Real-world data and measured channel

The real channel is an indoor 64-antenna RENEW massive MIMO channel [58]. We set up 8 users in the vicinity of the receivers. We use the RENEWLab real-time software[¶] to collect datasets in all user configurations and the 802.11 LTS signal as the pilot symbol which is used for channel estimation and beamforming matrix calculation. The format of the data OFDM symbols also follows the 802.11 standard where the FFT size is 64. Out of the 64 subcarriers, 48 subcarriers carry data, 4 are pilot subcarriers, and the remaining are null subcarriers. The pilot subcarriers are used for residual phase offset correction. We use QPSK as the modulation scheme for our experiments.

The first experiment is shown in Fig. 3.4a for a QPSK constellation. In this case, we generate synthetic observations using a batch of 800 channels. We compare with MMSE and ML. We see that our proposed Langevin detector outperforms MMSE and achieves a performance similar to ML.

The second experiment was carried out with real observations, with an SNR$\approx$20 dB (this is an average value across all the dataset). For this experiment, we change the

[¶]The implementation can be found at `https://github.com/renew-wireless/RENEWLab`

hyper-parameters of our proposed detector based on an empirical analysis. In particular, we set $M = 2$ trajectories and we increase the number of noise levels to 25 and set $\sigma_L = 0.001$. In Figs. 3.4c and 3.4d we show for 200 observations (time frames) for two users, considering one OFDM symbol and one subcarrier. We consider the 8 users and calculate the number of errors for each OFDM symbol and one subcarrier. We compute the histogram of this result, which is shown in Fig. 3.4b, where each point corresponds to the number of error in 200×8 observations per OFDM symbol and subcarrier. From a total number of 960k observations, 12880 symbols are incorrectly detected by Langevin, while 15212 by MMSE, achieving a SER of 0.016 and 0.02 respectively. Overall, compared to the synthetic data, the performance of both methods is worse. Moreover, although the gap in performance reduces between both methods, notice that for high SNR both perform similarly.

3.5.4 Performance for multiple modulation schemes simultaneously

In this subsection we consider the case of users transmitting with different modulation schemes simultaneously. In particular, we assume two different modulations. Formally, given two constellation sets $\mathcal{X}_1$ and $\mathcal{X}_2$, we define the following forward model

$$\mathbf{y} = \mathbf{H} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} + \mathbf{z}, \tag{3.25}$$

where $\mathbf{y} \in \mathbb{C}^{N_r}$, $\mathbf{H} \in \mathbb{C}^{N_r \times N_t}$ and $\mathbf{x}_1 \in \mathcal{X}_1^{N_u/2}, \mathbf{x}_2 \in \mathcal{X}_2^{N_u/2}$ are two vectors of transmitted symbols corresponding to one of the two modulations. For the detection, we assume that we know the modulation scheme used by each user. We use the Kronecker correlated channel model 3.24 and the same simulation setting as in Section 3.5.2.

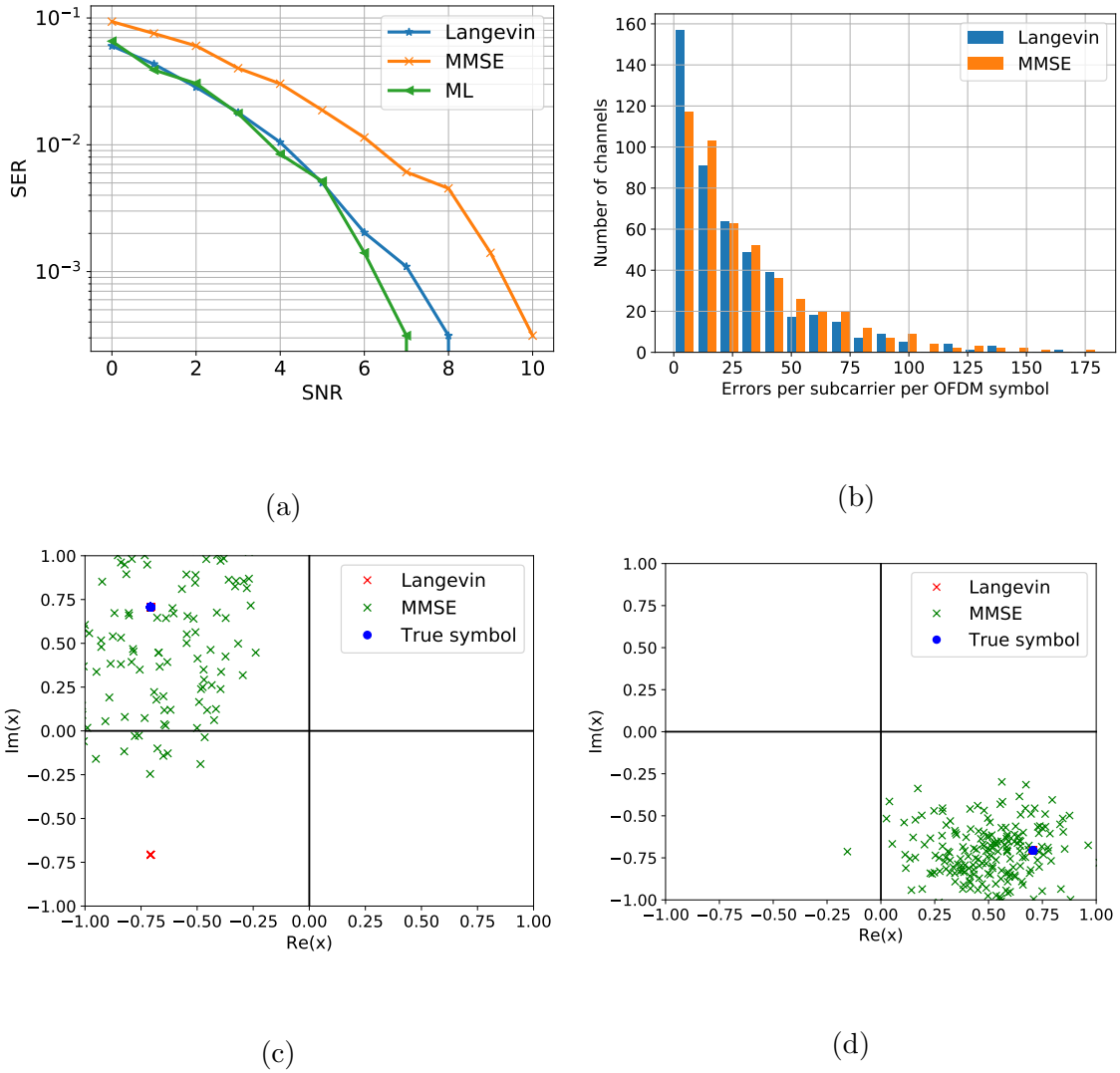


Figure 3.4 : (a) SER as a function of SNR for our Langevin method using the real channel defined in Section 3.5.3 and with synthetic data. The modulation is 4-QAM. (b) Histogram of the number of errors per subcarrier and OFDM symbol for each of the 200 time and frame and 8 users. (c)-(d) Symbol's estimation for two users and 200 different observations. We consider one OFDM symbol and one subcarrier for both users, and send the same symbol in the 200 time frames.

The result is shown in Fig. 3.5a for $\mathcal{X}_1$ = 16-QAM and $\mathcal{X}_2$ = 64-QAM. We compare only with the MMSE, as it is the only low-complexity detector that can handle different modulations simultaneously, and consider two different SNR ranges, one for each modulation. The figure reveals that our method outperforms the MMSE detector for both modulations. However, we observe that the SER performance for the lowest modulation decreases compared to the result in Fig. 3.3b. On the other hand, the performance for the higher modulation is consistently superior to MMSE for all SNR, remaining a similar behavior to the case of a unique modulation. Thus, we can conclude that when using different modulation schemes, Langevin consistently outperforms MMSE for all the modulation, although the performance for the users transmitting with lower modulation slightly decreases compared to the case when all users are using the same modulation.

As of last observation, notice that although our experiment was carried out with two modulations, we can seamlessly go to more than two constellations.

3.5.5 Imperfect CSI

We evaluate the effect of channel estimation error (imperfect CSI) on the symbol detection performance. To simulate this scenario, we first generate a batch of channels $\mathbf{H}$ from the Kronecker correlated channel model (3.24) and observations $\mathbf{y}$ using this set of channels; for the learning-based methods, we train using the true channels as well as the observations. Then, we run all the detectors using a perturbed channel

$$\hat{\mathbf{H}} = \mathbf{H} + \mathbf{E}, \tag{3.26}$$

where $\mathbf{E} \sim \mathcal{CN}(0, \sigma_e^2\mathbf{I})$ represents the estimation error. We consider an SNR = 16dB for the measurement noise (3.23), and $\sigma_e \in \{0.002, 0.02, 0.034, 0.06, 0.1\}$ for the

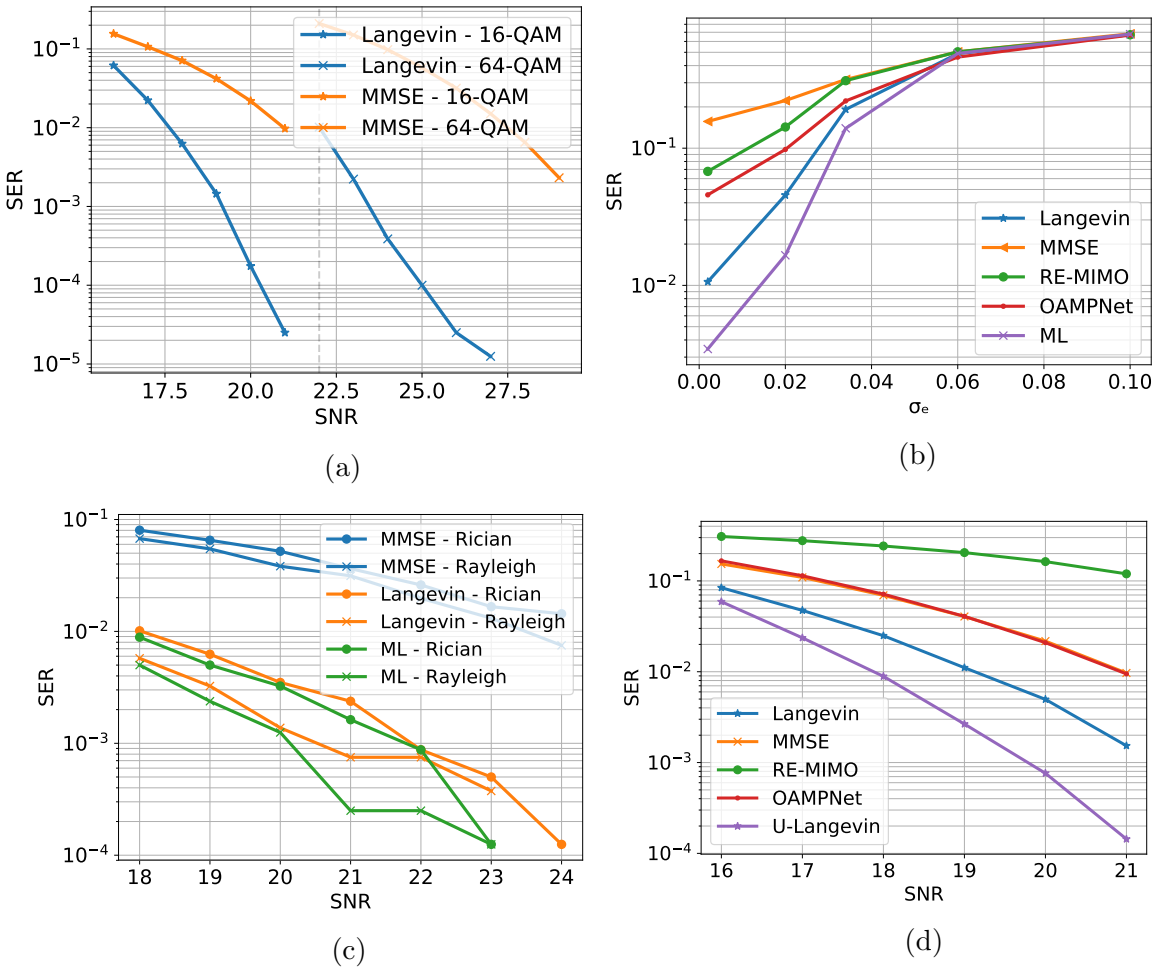


Figure 3.5 : Performance analysis of our proposed method evaluated in a Kronecker correlated channel model as in (3.24). (a) SER as a function of SNR for our Langevin method for users transmitting with two 16-QAM and 64-QAM simultaneously. (b) Comparison of our proposed MIMO detectors with other baseline detector when there is imperfect channel state information. We consider SNR = 16dB and $\sigma_e \in \{0.002, 0.02, 0.034, 0.06, 0.1\}$. (c) Comparison of our proposed MIMO detector with the MMSE and ML detectors for channels that belong to the Rician and Rayleigh models. (d) SER as a function of SNR when there is noise variance uncertainty.

variance of the estimation error. The SER as a function of σ_e is shown if Fig. 3.5b. We observe that for low variance, our method still outperforms the other detectors. As the error variance grows, all the methods collapse to the same performance.

3.5.6 Line-of-sight (LOS) and non-LOS (NLOS) channels

We analyze the SER when we consider a spatially correlated Rician fading model [59], which is composed by a deterministic part, known as the LOS component, and a stochastic part that corresponds to the NLOS component. Formally, this model is described as follows: given a user u, the channel between u and the BS antennas is given by $\mathbf{h}_u \sim \mathcal{CN}(\bar{\mathbf{h}}_u, \mathbf{R})$ where $\bar{\mathbf{h}}_u$ corresponds to the LOS component and $\mathbf{R}$ is the positive semi-definite covariance matrix describing the spatial correlation of the NLOS component. To get samples from this model, we generate each component separately: first the NLOS is sampled from $\mathcal{CN}(\mathbf{0}, \mathbf{R})$ and then we add the LOS component $\bar{\mathbf{h}}_u$ by choosing some users u randomly. Notice that the NLOS component alone corresponds to a Rayleigh fading model. We consider one BS with $N_r = 32$ and $N_u = 16$, and we use the code in [59] to generate 500 channels realizations. The result is show in Fig. 3.5c. We compare Langevin with MMSE and ML and we observe that in both settings the proposed Langevin method performs similar to ML.

3.5.7 Robustness to noise variance uncertainty

The Kronecker correlated channel model 3.24 is used and the same simulation setting as in Section 3.5.2. For both OAMPNet and RE-MIMO we trained the model assuming that we know the variance, and then at testing time, we replace the noise variance with an estimation, given by $\hat{\sigma} = \eta\sigma$. In particular, we consider $\eta = 0.15$ (8dB). In this case, we generate $M = 10$ samples.

The result is shown in Fig. 3.5d. We observe that both learning-based methods are highly-dependent on a good estimation of the noise variance. On the other hand, the performance of MMSE remains invariant. Compared to Langevin without learning, the proposed U-Langevin outperforms under this setting of noise uncertainty. Therefore, the U-Langevin is a more suitable detector in scenarios with noise variance uncertainty as it enhances performance while maintaining the same complexity. However, in the case of known noise variance, the best option is the Langevin detector without learning as we have closed-form expression and we have shown that it achieves state-of-the-art performance.

Chapter 4

Joint Massive MIMO data detection and channel estimation based on Lagevin diffusion

In this chapter, we address the problem of joint data detection and channel estimation in massive MIMO systems. In Chapter 3.1, we introduced a MIMO detector, which relies on perfect channel state information. This assumption is not true in general, as we do not have access to the CSI. Therefore, in this chapter we seek a method to estimate jointly the transmitted data the channel state information based on noisy observations and pre-defined signals known at the transmitter and receiver side (known as pilot signals). Similar to the case of MIMO detection, the direct solution, namely maximum-likelihood (ML), is NP-hard. In this context, I propose in this chapter an algorithm to solve this problem based on again the Langevin diffusion introduced in Section 2.1. The organization of this chapter is as follows: in Section 4.1, we present the system model and formulate the joint massive MIMO data detection and channel estimation problem. Then, in Section 3.3, we provide a detailed description of our proposed algorithm. We illustrate results through numerical experimentation, demonstrating the superior performance of our detector in comparison with other benchmark methods in Section 3.5.

4.1 System model and problem formulation

For the case of joint symbol detection and channel estimation, we slightly change the notation. We now consider N_u single-antenna transmitters or UEs that transmit

pilots and data to a receiving base station equipped with N_r antennas, under a block-fading scenario with a coherence time of $K = P + D$ time slots; P time slots are reserved for pilots and D time slots are used for payload data. Under this scenario, the forward model for the MIMO system is defined as

$$\mathbf{Y} = \mathbf{H}\mathbf{X} + \mathbf{Z}, \tag{4.1}$$

where $\mathbf{H} \in \mathbb{C}^{N_r \times N_u}$ is the channel matrix, each $\{[\mathbf{Z}]_{:j}\}_{j=1}^{K} \sim \mathcal{CN}(\mathbf{0}, \sigma_0^2 \mathbf{I}_{N_r})$ is a vector of complex circular Gaussian noise, $\mathbf{X} = [\mathbf{X}_P, \mathbf{X}_D] \in \mathcal{X}^{N_u \times K}$ is the vector of transmitted data, where $\mathbf{X}_P \in \mathcal{X}^{N_u \times P}$ corresponds to the pilots while $\mathbf{X}_D \in \mathcal{X}^{N_u \times D}$ is the data, and $\mathcal{X}$ is a finite set of constellation points, and $\mathbf{Y} \in \mathbb{C}^{N_r \times K}$ is the received vector. Under this configuration, the joint symbol detection and channel estimation can be stated as follows.

Problem 2 Given observations $\mathbf{Y}$ and pilots $\mathbf{X}_P$ following (4.1), find the MAP estimate of both $\mathbf{X}_D$ and $\mathbf{H}$.

Given that $\mathbf{Z}$ in (4.1) is a random variable, a natural way of solving Problem 1 is to search for $\{\mathbf{H}, \mathbf{X}_D\}$ that maximizes its *posterior* probability given the noisy observations $\mathbf{Y}$ and pilots $\mathbf{X}_P$. Hence, the Bayes' optimal decision rule can be written as

$$\begin{aligned} \{\hat{\mathbf{H}}_{\text{MAP}}, \hat{\mathbf{X}}_{D_{\text{MAP}}}\} &= \underset{\substack{\mathbf{X}_D \in \mathcal{X}^{N_u \times D}, \\ \mathbf{H} \in \mathbb{C}^{N_r \times N_u}}}{\operatorname{argmax}} \; p(\mathbf{X}_D, \mathbf{H} | \mathbf{Y}, \mathbf{X}_P) \\ &= \underset{\substack{\mathbf{X}_D \in \mathcal{X}^{N_u \times D}, \\ \mathbf{H} \in \mathbb{C}^{N_r \times N_u}}}{\operatorname{argmax}} \; p_{\mathbf{Z}}(\mathbf{Y} - \mathbf{H}\mathbf{X}) p(\mathbf{X}) p(\mathbf{H}). \end{aligned} \tag{4.2}$$

The problem in (4.3) is known as MAP-JED. Since we assume that the symbols' prior distribution is uniform among the constellation elements and the measurement noise $\mathbf{Z}$ is Gaussian, the MAP-JED formulation boils down to the following optimization

problem

$$\{\hat{\mathbf{H}}_{\text{MAP}}, \hat{\mathbf{X}}_{D_{\text{MAP}}}\} = \underset{\substack{\mathbf{X}_D \in \mathcal{X}^{N_u \times D}, \\ \mathbf{H} \in \mathbb{C}^{N_r \times N_u}}}{\operatorname{argmin}} ||\mathbf{Y} - \mathbf{H}\mathbf{X}||_2^2 - \log p(\mathbf{H}), \tag{4.3}$$

This problem is also NP-hard due to the discrete nature of the finite constellation constraint $\mathbf{X}_D \in \mathcal{X}^{N_u}$ and the non-convexity of the product between decision variables.

4.2 Joint sampling based on Langevin diffusion.

Similar to the method proposed in Chapter 3.1, our goal is to solve Problem 1 via ULA (2.6) by sampling from the joint posterior (4.2). Hence, we must adapt the framework from Section 2.1 for several reasons. First, *we are interested in sampling from the joint posterior* $p(\mathbf{X}_D, \mathbf{H}|\mathbf{Y}, \mathbf{X}_P)$*, not just* $p(\mathbf{X})$*.* This necessitates computing the score of the joint posterior, specifically gradients with respect to $\mathbf{X}_D$ and $\mathbf{H}$, expressed as:

$$\begin{aligned} \nabla_{\mathbf{X}_D} \log p(\mathbf{X}_D, \mathbf{H}|\mathbf{Y}, \mathbf{X}_P) &= \nabla_{\mathbf{X}_D} \log p(\mathbf{Y}, \mathbf{X}_P|\mathbf{H}, \mathbf{X}_D) + \nabla_{\mathbf{X}_D} \log p(\mathbf{X}_D), \\ \nabla_{\mathbf{H}} \log p(\mathbf{X}_D, \mathbf{H}|\mathbf{Y}, \mathbf{X}_P) &= \nabla_{\mathbf{H}} \log p(\mathbf{Y}, \mathbf{X}_P|\mathbf{H}, \mathbf{X}_D) + \nabla_{\mathbf{H}} \log p(\mathbf{H}), \end{aligned} \tag{4.4}$$

where $\nabla_{\mathbf{X}_D} \log p(\mathbf{Y}, \mathbf{X}_P|\mathbf{H}, \mathbf{X}_D)$ and $\nabla_{\mathbf{X}_D} \log p(\mathbf{X}_D)$ are the score functions of the likelihood and prior with respect to $\mathbf{X}_D$ respectively, while $\nabla_{\mathbf{H}} \log p(\mathbf{Y}, \mathbf{X}_P|\mathbf{H}, \mathbf{X}_D)$ and $\nabla_{\mathbf{H}} \log p(\mathbf{H})$ are the scores with respect to $\mathbf{H}$. Second, *computing the gradient for the prior of* $\mathbf{X}_D$ *is challenging due to its discrete nature.* To address this, we propose an *annealing process*, creating a continuous approximation of $\mathbf{X}_D$. We define a sequence of noise levels $\{\sigma_{l,\mathbf{X}}\}_{l=1}^{L}$, with decreasing values so that $\sigma_{L,\mathbf{X}} \approx 0$. Then, at each level, we introduce a perturbed version of the true symbols $\mathbf{X}_D$

$$\tilde{\mathbf{X}}_{D,l} = \mathbf{X}_D + \mathbf{N}_{\mathbf{X},l}, \tag{4.5}$$

where $\mathbf{N}_{\mathbf{X},l} \sim \mathcal{CN}(0, \sigma_{l,\mathbf{X}}^2 \mathbf{I})$. Hence, we utilize the perturbed symbol set $\tilde{\mathbf{X}}_D$ instead of the true variable $\mathbf{X}_D$, enabling the definition of a continuous score prior. Therefore, as noise levels decrease, $\tilde{\mathbf{X}}_D$ progressively converges to $\mathbf{X}_D$. Although this is similar to the concept introduced in Section 2.1, the ultimate goal is different, as we aim to create a continuous approximation of the discrete variable $\tilde{\mathbf{X}}_D$. Furthermore, we *lack a closed-form expression for the score of* $\mathbf{H}$*, prompting the use of a score network parameterization and training via denoising score matching*, as presented in Section 2.1.

In a nutshell, the algorithm follows these steps: initialization of $\tilde{\mathbf{X}}_{D,0}$ and $\tilde{\mathbf{H}}_0$ randomly. After that, it follows the score function of the joint posterior density of perturbed variables, starting with high $\sigma_{1,\mathbf{X}}$ and $\sigma_{1,\mathbf{H}}$, progressively reducing to $\sigma_{L,\mathbf{X}} \approx \sigma_{L,\mathbf{H}} \approx 0$. At early noise levels, the likelihood term directs the dynamics toward an estimate mainly driven by the measurements, while in later noise levels, the prior refines the estimate, as explained further in [29]. Annealing benefits are threefold: it is used to train the score network via score-matching, it enhances dynamic mixing, and it allows for discrete-to-continuous variable approximation. Consequently, we apply annealing to both $\mathbf{H}$ and $\mathbf{X}_D$ to harness the first two benefits for the former and the last two for the latter. With this high-level understanding, we now detail the terms in (3.8) and provide a step-by-step algorithm description.

4.2.1 Score functions.

The score functions are computed with respect to the perturbed variables. Thus, our model for running the dynamic in (2.6) is described by the following joint posterior distribution

$$p(\tilde{\mathbf{X}}_{D,l}, \tilde{\mathbf{H}}_l | \mathbf{Y}, \mathbf{X}_P) \propto p(\mathbf{Y}, \mathbf{X}_P | \tilde{\mathbf{X}}_{D,l}, \tilde{\mathbf{H}}_l) p(\tilde{\mathbf{X}}_{D,l}) p(\tilde{\mathbf{H}}_l). \tag{4.6}$$

Hence, we need to compute four score functions (4.4), two corresponding to the score of the likelihood terms, and two for the score prior.

i) Score of the likelihood term of $\tilde{\mathbf{X}}_{D,l}$: Under the new model in (4.6), the score of the likelihood term for $\tilde{\mathbf{X}}_{D,l}$ is given by $p(\mathbf{Y}|\tilde{\mathbf{X}}_l, \tilde{\mathbf{H}}_l, \mathbf{X}_P) = p(\mathbf{Z} - \tilde{\mathbf{H}}_l \mathbf{N}_{\mathbf{X},l}|\tilde{\mathbf{X}}_{D,l})$, which is not Gaussian: although $p(\mathbf{N}_{\mathbf{X},l})$ is a Gaussian distribution, after conditioning on the perturbed variable the conditional distribution $p(\mathbf{N}_{\mathbf{X},l}|\tilde{\mathbf{X}}_{D,l})$ is no longer Gaussian. To circumvent this, we consider the following *approximation*. First, notice that when $\sigma_{l,\mathbf{X}}$ is small, then $\nabla_{\tilde{\mathbf{X}}_D} \log p(\mathbf{Y}|\tilde{\mathbf{X}}_{D,l}, \tilde{\mathbf{H}}_l, \mathbf{X}_P) \approx \nabla_{\mathbf{X}_D} \log p(\mathbf{Y}_D|\mathbf{X}_D, \tilde{\mathbf{H}}_l) = \frac{\tilde{\mathbf{H}}_l^{\mathrm{H}}(\mathbf{Y}_D - \tilde{\mathbf{H}}_l \mathbf{X}_D)}{\sigma_0^2}$, with $\tilde{\mathbf{H}}_l^{\mathrm{H}}$ denoting the conjugate transpose of $\mathbf{H}$; this is only correct when $\sigma_{l,\mathbf{X}} \to 0$ (recall that for $\sigma_{l,\mathbf{X}} = 0$, the gradient is not defined). Second, given that the approximation is not valid for high $\sigma_{l,\mathbf{X}}$, we add a correction term as follows

$$\nabla_{\tilde{\mathbf{X}}_D} \log p(\mathbf{Y}|\tilde{\mathbf{X}}_l, \tilde{\mathbf{H}}_l, \mathbf{X}_P) \approx \tilde{\mathbf{H}}_l^{\mathrm{H}}(\sigma_0^2 \mathbf{I} + \sigma_{l,\mathbf{X}}^2 \tilde{\mathbf{H}}_l^{\mathrm{H}} \tilde{\mathbf{H}}_l)^{-1}(\mathbf{Y}_D - \tilde{\mathbf{H}}_l \tilde{\mathbf{X}}_D). \tag{4.7}$$

Intuitively, the approximation in (4.7) assumes that the annealing and measurement noise are independent. Although this approximation entails a worse symbol error rate (SER) than the one introduced in [29], it provides a good approximation that allows one to reduce the computational burden of the algorithm.

ii) Score of the likelihood term of $\tilde{\mathbf{H}}_l$*:* Similar to the approximation in (4.7), we approximate the score of likelihood of $\tilde{\mathbf{H}}_l$ as follows

$$\nabla_{\tilde{\mathbf{H}}_l} \log p(\mathbf{Y}|\tilde{\mathbf{X}}_l, \tilde{\mathbf{H}}_l, \mathbf{X}_P) \approx \frac{(\mathbf{Y} - \tilde{\mathbf{H}}_l[\tilde{\mathbf{X}}_D, \mathbf{X}_P])[\tilde{\mathbf{X}}_D, \mathbf{X}_P]^{\mathrm{H}}}{\sigma_0^2 + \sigma_{l,\mathbf{H}}^2} \tag{4.8}$$

Notice that this term considers both the pilots and the data in contrast to the score of the likelihood of $\tilde{\mathbf{X}}_D$ in (4.7)

iii) Score of the annealed prior of $\tilde{\mathbf{X}}_{D,l}$*:* The score function can be related to the

MMSE denoiser through Tweedie's identity [47] as follows

$$\nabla_{\tilde{\mathbf{X}}_{D,l}} \log p(\tilde{\mathbf{X}}_{D,l}) = \frac{\mathbb{E}_{\sigma_{l,\mathbf{X}}}[\mathbf{X}_D|\tilde{\mathbf{X}}_{D,l}] - \tilde{\mathbf{X}}_{D,l}}{\sigma^2_{l,\mathbf{X}}}. \tag{4.9}$$

In particular, the conditional expectation can be calculated elementwise as

$$\mathbb{E}_{\sigma_{l,\mathbf{X}}}[x_j|[\tilde{\mathbf{X}}_{D,l}]_{j,p}] = \frac{1}{Z}\sum_{x_k \in \mathcal{X}} x_k \exp\left(\frac{-||[\tilde{\mathbf{X}}_{D,l}]_{j,p} - x_k||^2}{2\sigma^2_{l,\mathbf{X}}}\right), \tag{4.10}$$

where $Z = \sum_{x_k \in \mathcal{X}} \exp\left(\frac{-||[\tilde{\mathbf{X}}_{D,l}]_{j,p} - x_k||^2}{2\sigma^2_{l,\mathbf{X}}}\right)$ and $j = 1, \cdots, N_u$.

iv) Score of the annealed prior of $\tilde{\mathbf{H}}_l$*:* Finally, the score prior for $\tilde{\mathbf{H}}_l$ is parameterized by a score network, and trained using denoising score matching, which was introduced in Section 2.1. Hence,

$$\nabla_{\tilde{\mathbf{H}}_l} \log p(\tilde{\mathbf{H}}_l) = \mathbf{s}_{\boldsymbol{\theta}}(\tilde{\mathbf{H}}_l, \sigma_{l,\mathbf{H}}), \tag{4.11}$$

where the network $\mathbf{s}_{\boldsymbol{\theta}}(.)$ is trained by minizing the loss function in (2.12).

4.2.2 Langevin-based joint MIMO data detection and channel estimation algorithm.

The algorithm to generate estimates $\hat{\mathbf{H}}$ and $\hat{\mathbf{X}}_D$ by sampling from the (approximate) joint posterior $p(\mathbf{X}_D, \mathbf{H}|\mathbf{Y}, \mathbf{X}_P)$ is shown in Algorithm 2. From the update steps, we observe that the main advantage of performing joint sampling is to reuse data symbols as pilots, which entails two benefits. First, it allows sending more data with the same data rate and maintaining a given system performance. Second, it reduces the channel estimation error as long as the estimation of the symbols is good. Hence, one might expect to see an improvement at higher SNRs. Finally, from an implementation perspective, the selection of the hyperparameters – the number of levels of noise as well as the step sizes – is key for the stability of the algorithm.

Algorithm 2 Annealed Langevin for JED-MAP

Require: $T, L, \{\sigma_{l,\mathbf{X}}, \sigma_{l,\mathbf{H}}\}_{l=1}^{L}, \epsilon_{\mathbf{X}}, \epsilon_{\mathbf{H}}, \sigma_0, \mathbf{Y}, \mathbf{X}_P, \tau_{\mathbf{X}}, \tau_{\mathbf{H}}$

Initialize $\tilde{\mathbf{X}}_{D_{t=0,l=1}}, \tilde{\mathbf{H}}_{t=0,l=1}$ randomly

for $l = 1$ to L **do**

Compute $\epsilon_{l,\mathbf{X}} = \epsilon_{\mathbf{X}} \cdot \left(\frac{\sigma_{l,\mathbf{X}}}{\sigma_{L,\mathbf{X}}}\right)^2, \epsilon_{l,\mathbf{H}} = \epsilon_{\mathbf{H}} \cdot \left(\frac{\sigma_{l,\mathbf{H}}}{\sigma_{L,\mathbf{H}}}\right)^2$

for $k = 0$ to $T-1$ **do**

Draw $\mathbf{Z}_k \sim \mathcal{N}(0, \mathbf{I})$

Compute $\nabla_{\tilde{\mathbf{X}}_{D_{k,l}}} \log p(\mathbf{Y}|\tilde{\mathbf{X}}_{D_{k,l}}, \tilde{\mathbf{H}}_{k,l}, \mathbf{X}_P)$ as in (4.7)

Compute $\nabla_{\tilde{\mathbf{X}}_{D_{k,l}}} \log p(\tilde{\mathbf{X}}_{D_{k,l}})$ as in (4.9)

Compute $\nabla_{\tilde{\mathbf{X}}_{D_{k,l}}} \log p(\tilde{\mathbf{X}}_{D_{k,l}}, \tilde{\mathbf{H}}_{k,l}|\mathbf{Y}, \mathbf{X}_P)$ as in (4.4)

$\tilde{\mathbf{X}}_{D_{k+1,l}} = \tilde{\mathbf{X}}_{D_{k,l}} + \epsilon_{l,\mathbf{X}} \nabla_{\mathbf{X}_{D_{k,l}}} \log p(\tilde{\mathbf{X}}_{D_{k,l}}, \tilde{\mathbf{H}}_{k,l}|\mathbf{Y}, \mathbf{X}_P)$
$+\sqrt{2\epsilon_{l,\mathbf{X}}\, \tau_{\mathbf{X}}}\, \mathbf{Z}_k$

Compute $\nabla_{\tilde{\mathbf{H}}_{k,l}} \log p(\mathbf{Y}|\tilde{\mathbf{X}}_{D_{k+1,l}}, \tilde{\mathbf{H}}_{k,l}, \mathbf{X}_P)$ as in (4.8)

Compute $\nabla_{\tilde{\mathbf{H}}_l} \log p(\tilde{\mathbf{H}}_l)$ as in (4.11)

Compute $\nabla_{\tilde{\mathbf{H}}_{k,l}} \log p(\tilde{\mathbf{X}}_{D_{k+1,l}}, \tilde{\mathbf{H}}_{k,l}|\mathbf{Y}, \mathbf{X}_P)$ as in (4.4)

$\tilde{\mathbf{H}}_{k+1,l} = \tilde{\mathbf{H}}_{k,l} + \epsilon_{l,\mathbf{H}} \nabla_{\tilde{\mathbf{H}}_{k,l}} \log p(\tilde{\mathbf{X}}_{D_{k+1,l}}, \tilde{\mathbf{H}}_{k,l}|\mathbf{Y}, \mathbf{X}_P)$
$+\sqrt{2\epsilon_{l,\mathbf{H}}\, \tau_{\mathbf{H}}}\, \mathbf{Z}_k$

end for

end for

return $\hat{\mathbf{X}}_D = \tilde{\mathbf{X}}_{D_{T,L}}, \hat{\mathbf{H}} = \tilde{\mathbf{H}}_{T,L}$

Computational complexity. The complexity of Algorithm 2 is the combination of the complexities of each dynamic. The dynamic with respect to the variable $\mathbf{X}_D$ yields a complexity where the main bottleneck is the matrix inversion in (4.7). Hence, the complexity of computing the guidance term and the score prior is $\mathcal{O}(N_u^3 + KN_u)$. A deeper analysis of the complexity of this method can be found in [29]. For the case of $\mathbf{H}$, the heavier computation is carried by the score network. An analysis of the computation complexity of this method can be found in [31].

4.3 Numerical Experiments

In this section, we evaluate the performance of our method. * We begin by introducing the channel model and simulation setup. The first experiment assesses channel estimation error versus signal-to-noise ratio (SNR), varying the number of symbols $\mathbf{X}_D$ with a fixed pilot count, comparing against the case with no symbol reuse as pilots. In the second experiment, we consider the same setting as experiment one and evaluate the SER as a function of SNR. Finally, we benchmark our approach against baseline methods in channel estimation.

Channel model and simulation settings. We consider a channel model that is representative of the 3GPP 3D MIMO channel model [56], as implemented in the QuaDRiGa channel simulator [57]. We consider a base station with an 8×8 half-wavelength spacing ($N_r = 64$), single-polarization antenna array at a height of 20 m. We assume that the BS covers a sector of radius 500 m and $N_u = 32$ single-polarization omni-directional antennas users are dropped randomly in the coverage area. Moreover, users are NLOS and indoors. The carrier frequency is 3.5 GHz, and each subcarrier has a 100 MHz bandwidth. The spacing within subcarriers is 30 kHz. For the score network, we use the parameterization proposed in [26], and minimize the loss in (3.22). We consider a training dataset with 7000 channels, and a validation set of 2500 channels. We evaluate the performance in a batch size of 50 matrices. For the pilots, we generate P QPSK symbols. The parameters of the Algorithm 2 are $L = 2311, T = 3, \epsilon_{\mathbf{H}} = 1 \times 10^{-10}, \tau_{\mathbf{H}} = 1 \times 10^{-3}$ and $[\sigma_{1,\mathbf{H}}, \sigma_{L,\mathbf{H}}] = [30, 0.001]$; lastly, for the dynamic with respect to $\mathbf{X}_D$, we use two different values depending on the SNR: for low SNR (5 dB-15 dB) we consider $\epsilon_{\mathbf{X}} = 1 \times 10^{-4}, \tau_{\mathbf{X}} = 0.5$ and

*Code for reproducing experiments is available at `https://github.com/nzilberstein/Langevin-joint-channel`

$[\sigma_{1,\mathbf{X}}, \sigma_{L,\mathbf{X}}] = [0.6, 0.01]$, while for all other SNR values we fixed $\epsilon_{\mathbf{X}} = 4 \times 10^{-5}, \tau_{\mathbf{X}} = 0.1$ and $[\sigma_{1,\mathbf{X}}, \sigma_{L,\mathbf{X}}] = [0.8, 0.01]$. Lastly, the noise variance σ_0^2 changes for different levels of SNR. If computationally limiting, L can be reduced by considering higher-order Langevin dynamics; see [60] for an analysis of the case of channel sampling.

NMSE performance for varying number of symbols. In this first experiment, we analyze the performance of our proposed algorithm for channel estimation with different values for D, i.e., symbols to reuse as pilots, and fixing the value of $P = 30$. The comparison is shown in Fig. 4.1, where we named *single Langevin* the channel sampling for $D = 0$, proposed in [31]. As expected, performance improves with an increasing number of data symbols for pilot reuse. Notably, when $D > 30$, our method excels for SNRs above 20 dB, and for $D > 50$, it outperforms single Langevin across all SNRs. Overall, superior performance is evident for SNRs higher than 20 dB.

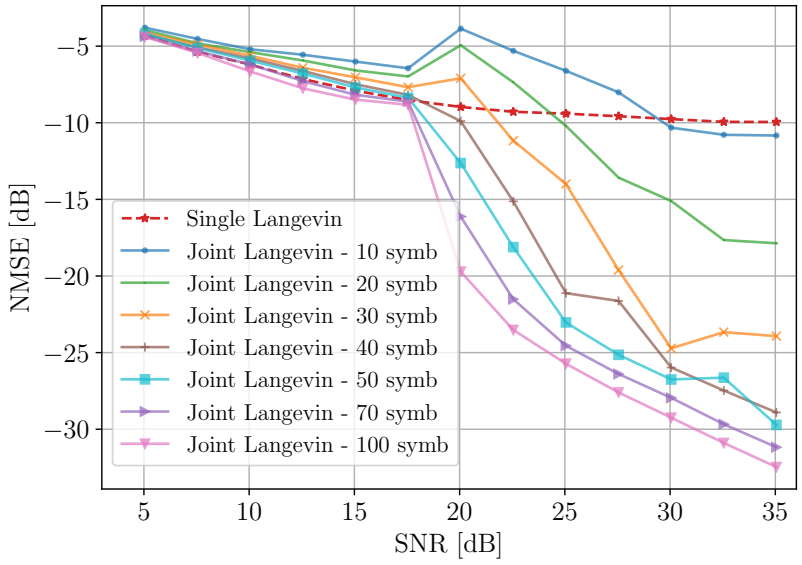


Figure 4.1 : Performance analysis of our proposed method as a function of SNR for a 3GPP channel model. NMSE for a fixed number of pilot symbols $P = 30$ and different number of data symbols, $D = \{10, 20, 30, 40, 50, 70, 100\}$

SER performance for varying number of symbols. In the same experimental conditions as before (where $P = 30$), we examine the SER as a function of SNR, with varying values of D. The results are presented in Fig. 4.2. Similar to the behavior observed in channel estimation error, we notice an improvement in SER as D increases. Notably, when $D > 30$, our proposed algorithm outperforms the single Langevin method across all SNRs. These findings suggest that the limitation in NMSE performance in our algorithm stems from symbol sampling performance. In high SNR scenarios, accurate estimation of $\mathbf{X}_D$ leads to precise joint variable estimation. However, in lower SNR situations, there is not a significant advantage in considering symbol reuse.

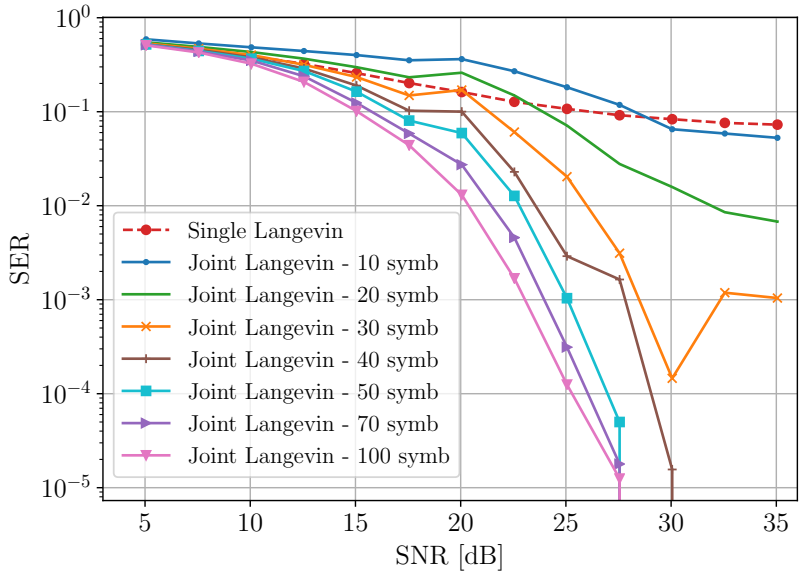


Figure 4.2 : Performance analysis of our proposed method as a function of SNR for a 3GPP channel model. SER for a fixed number of pilot symbols $P = 30$ and different number of data symbols, $D = \{10, 20, 30, 40, 50, 70, 100\}$

Performance comparison with baseline methods. In our third experiment, we compare our proposed algorithm to several baseline methods: LASSO [61], fsAD [62],

L-MMSE [63], L-DAMP [64], and Langevin using only pilots [31]. We set $P = 30$ for all baseline algorithms and $D = 50$ for our method. Results in Fig. 4.3 reveal that our approach consistently outperforms all baselines across all SNRs. Particularly noteworthy is its superior performance, exceeding other methods by several orders of magnitude for SNRs above 15 dB.

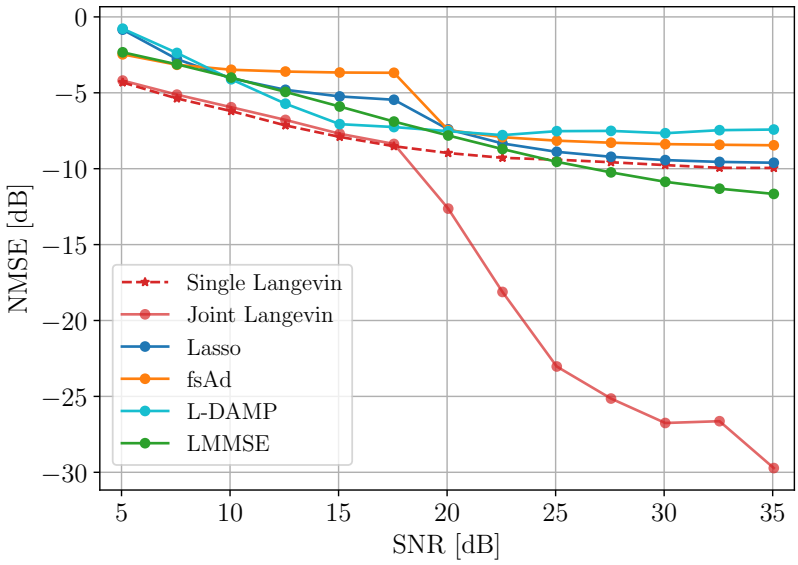


Figure 4.3 : Comparison with different baseline methods, for $P = 30$ and $D = 50$.

Chapter 5

Conclusions

In this thesis, we proposed algorithms based on an annealed version of Langevin dynamics for solving two inverse problems in massive MIMO systems. Specifically, we considered the problem of MIMO detection, where we proposed a state-of-the-art method, and the and joint MIMO data detection and channel estimation. For both problems, the annealing process allows us to naturally incorporate the discrete nature of the constellation elements in the exploration of the search space by the relationship between the score of the prior and the denoiser; and to learn a meaningful representation of the prior information about the channels. Furthermore, for the case of MIMO detecion in scenarios where the noise variance is unknown, we relied on the algorithm unrolling framework to parameterize the score of the likelihood.

Revisiting the motivation question in Section 1, we demonstrated that incorporating prior knowledge of the underlying signal is key to develop efficient algorithms that achieve high performance. In particular, we show: 1) when we know the underlying structure of the signal, like in MIMO detection where we know that symbols are discrete, we can leverage to generate high-quality samples; 2) in the case when the underlying structure of the signal is unknown, we can learn the prior structure by training a score network. The combination of both techniques also enhances the sampling, as it was shown in the JED-MAP algorithm. We show by an extensive analysis and experiments that including this information yields state-of-the-art solutions.

The main limitation of the proposed methods is the running time and complexity,

in particular for fast-fading channels. To circumvent this, we explored alternative stochastic diffusion processes that work in non-asymptotic regimes. This allows to reduce the number of iterations while still sampling from the posterior distribution considering the discrete nature of the symbols. Future work includes deriving a noise-agnostic JED-MAP algorithm. We believe that using denoising score likelihood is an interesting direction to explore in this endeavor [65].

Bibliography

[1] S. Yang and L. Hanzo, "Fifty years of MIMO detection: The road to large-scale MIMOs," *IEEE Commun. Surveys Tut.*, vol. 17, no. 4, pp. 1941–1988, 2015.

[2] A. J. Paulraj, D. A. Gore, R. U. Nabar, and H. Bolcskei, "An overview of MIMO communications - a key to gigabit wireless," vol. 92, no. 2, pp. 198–218, 2004.

[3] K. B. Letaief, W. Chen, Y. Shi, J. Zhang, and Y.-J. A. Zhang, "The roadmap to 6G: AI empowered wireless networks," vol. 57, no. 8, pp. 84–90, 2019.

[4] A. D. Pia, S. S. Dey, and M. Molinaro, "Mixed-integer quadratic programming is in NP," *Mathematical Programming*, vol. 162, pp. 225–240, 2017.

[5] E. Balevi, A. Doshi, A. Jalal, A. Dimakis, and J. G. Andrews, "High dimensional channel estimation using deep generative networks," vol. 39, no. 1, pp. 18–30, 2020.

[6] J. G. Proakis, *Digital Communications 5th Edition*. McGraw Hill, 2007.

[7] A. Chockalingam and B. S. Rajan, *Large MIMO Systems*. Cambridge University Press, 2014.

[8] N. Samuel, T. Diskin, and A. Wiesel, "Deep MIMO detection," pp. 1–5, 2017.

[9] K. Pratik, B. D. Rao, and M. Welling, "RE-MIMO: Recurrent and permutation equivariant neural MIMO detection," vol. 69, pp. 459–473, 2021.

[10] H. He, C.-K. Wen, S. Jin, and G. Y. Li, "A model-driven deep learning network for MIMO detection," pp. 584–588, 2018.

[11] N. Zilberstein, C. Dick, R. Doost-Mohammady, A. Sabharwal, and S. Segarra, "Robust MIMO detection using hypernetworks with learned regularizers," *arXiv preprint arXiv:2110.07053*, 2021.

[12] M. Goutay, F. Ait Aoudia, and J. Hoydis, "Deep hypernetwork-based MIMO detection," pp. 1–5, 2020.

[13] O. Castaneda, T. Goldstein, and C. Studer, "VLSI designs for joint channel estimation and data detection in large SIMO wireless systems," *IEEE Trans. Circuits Syst. I Regul. Pap.*, vol. 65, no. 3, pp. 1120–1132, 2017.

[14] H. Song, T. Goldstein, X. You, C. Zhang, O. Tirkkonen, and C. Studer, "Joint channel estimation and data detection in cell-free massive MU-MIMO systems," vol. 21, no. 6, pp. 4068–4084, 2021.

[15] H. Song, X. You, C. Zhang, and C. Studer, "Soft-output joint channel estimation and data detection using deep unfolding," in *IEEE Inf. Theory. Workshop*, pp. 1–5, IEEE, 2021.

[16] B. Øksendal, *Stochastic differential equations*. Springer, 2003.

[17] G. A. Pavliotis, *Stochastic Processes and Applications: Diffusion Processes, the Fokker-Planck and Langevin Equations*. Springer, 2014.

[18] C. Robert and G. Casella, *Monte Carlo Statistical Method*. Springer, 1999.

[19] G. O. Roberts and R. L. Tweedie, "Exponential convergence of Langevin distributions and their discrete approximations," *Bernoulli*, vol. 2, pp. 341–363,

1996.

[20] M. Welling and Y. W. Teh, "Bayesian learning via stochastic gradient langevin dynamics," in *Intl. Conf. on Machine Learning (ICML)*, p. 681–688, 2011.

[21] A. S. Dalalyan and A. Karagulyan, "User-friendly guarantees for the langevin monte carlo with inaccurate gradient," *Stoch. Process. Their Appl.*, vol. 129, no. 12, pp. 5278–5311, 2019.

[22] A. Hyvärinen and P. Dayan, "Estimation of non-normalized statistical models by score matching.," *Journal of Machine Learning Research*, vol. 6, no. 4, 2005.

[23] P. Vincent, "A connection between score matching and denoising autoencoders," *Neural Computation*, vol. 23, no. 7, pp. 1661–1674, 2011.

[24] Y. Song and S. Ermon, "Generative modeling by estimating gradients of the data distribution," p. 11918–11930, 2019.

[25] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," in *Intl. Conf. Learn. Repr. (ICLR)*, 2021.

[26] Y. Song and S. Ermon, "Improved techniques for training score-based generative models," *arXiv preprint arXiv:2006.09011*, 2020.

[27] B. Kawar, M. Elad, S. Ermon, and J. Song, "Denoising diffusion restoration models," 2022.

[28] H. Chung, J. Kim, M. T. Mccann, M. L. Klasky, and J. C. Ye, "Diffusion posterior sampling for general noisy inverse problems," in *Intl. Conf. Learn. Repr. (ICLR)*, 2023.

[29] N. Zilberstein, C. Dick, R. Doost-Mohammady, A. Sabharwal, and S. Segarra, "Annealed Langevin dynamics for massive MIMO detection," vol. 22, no. 6, 2023 (Online Nov 2022).

[30] N. Zilberstein, C. Dick, R. Doost-Mohammady, A. Sabharwal, and S. Segarra, "Accelerated massive MIMO detector based on annealed underdamped Langevin dynamics," 2023.

[31] M. Arvinte and J. I. Tamir, "MIMO channel estimation using score-based generative models," vol. 22, no. 6, 2023 (Online Nov 2022).

[32] B. Levac, A. Jalal, and J. I. Tamir, "Accelerated motion correction for MRI using score-based generative models," *arXiv preprint arXiv:2211.00199*, 2022.

[33] H. Chung, J. Kim, S. Kim, and J. C. Ye, "Parallel diffusion models of operator and image for blind inverse problems," in *IEEE Conf. Comp. Vision Pattern Recognit. (CVPR)*, pp. 6059–6069, June 2023.

[34] M. Bayati and A. Montanari, "The dynamics of message passing on dense graphs, with applications to compressed sensing," vol. 57, no. 2, pp. 764–785, 2011.

[35] J. Ma and L. Ping, "Orthogonal AMP," *IEEE Access*, vol. 5, pp. 2020–2033, 2017.

[36] A. M. Tulino and S. Verdú, "Random matrix theory and wireless communications," *Found. Trends Commun. Inf. Theory.*, vol. 1, no. 1, pp. 1–182, 2004.

[37] P. W. Wolniansky, G. J. Foschini, G. D. Golden, and R. A. Valenzuela, "V-BLAST: an architecture for realizing very high data rates over the rich-scattering

wireless channel," in *Proc. URSI Int. Symp. Signals, Syst., Electron. Conf. Proc.*, pp. 295–300, 1998.

[38] A. Wiesel, Y. C. Eldar, and S. Shamai, "Semidefinite relaxation for detection of 16-QAM signaling in MIMO channels," vol. 12, no. 9, pp. 653–656, 2005.

[39] T. Datta, N. A. Kumar, A. Chockalingam, and B. S. Rajan, "A novel Monte-Carlo-sampling-based receiver for large-scale uplink multiuser MIMO systems," vol. 62, no. 7, pp. 3019–3038, 2013.

[40] N. M. Gowda, S. Krishnamurthy, and A. Belogolovy, "Metropolis-hastings random walk along the gradient descent direction for MIMO detection," in *IEEE Intl. Conf. on Comms.*, pp. 1–7, 2021.

[41] Z. Wu and H. Li, "Stochastic gradient Langevin dynamics for massive mimo detection," *IEEE Commun. Lett.*, pp. 1–1, 2022.

[42] N. Zilberstein, C. Dick, R. Doost-Mohammady, A. Sabharwal, and S. Segarra, "Detection by sampling: Massive MIMO detector based on Langevin dynamics," *arXiv preprint arXiv:2202.12199*, 2022.

[43] M. Khani, M. Alizadeh, J. Hoydis, and P. Fleming, "Adaptive neural signal detection for massive MIMO," vol. 19, no. 8, pp. 5635–5648, 2020.

[44] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imaging Sci.*, vol. 2, no. 1, pp. 183–202, 2009.

[45] P. Putzky and M. Welling, "Recurrent inference machines for solving inverse problems," *arXiv preprint arXiv:1706.04008*, 2017.

[46] B. Kawar, G. Vaksman, and M. Elad, "SNIPS: Solving noisy inverse problems stochastically," vol. 34, pp. 21757–21769, 2021.

[47] B. Efron, "Tweedie's formula and selection bias," *Journal of the American Stat. Association*, vol. 106, no. 496, pp. 1602–1614, 2011.

[48] V. Monga, Y. Li, and Y. C. Eldar, "Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing," *IEEE Signal Process. Mag.*, vol. 38, no. 2, pp. 18–44, 2021.

[49] A. Chowdhury, G. Verma, C. Rao, A. Swami, and S. Segarra, "Unfolding WMMSE using graph neural networks for efficient power allocation," vol. 20, no. 9, pp. 6004–6017, 2021.

[50] N. Farsad, N. Shlezinger, A. J. Goldsmith, and Y. C. Eldar, "Data-driven symbol detection via model-based machine learning," in *IEEE Stat. Signal Process. Workshop (SSP)*, pp. 571–575, 2021.

[51] K. He, L. He, L. Fan, Y. Deng, G. K. Karagiannidis, and A. Nallanathan, "Learning-based signal detection for mimo systems with unknown noise statistics," vol. 69, no. 5, pp. 3025–3038, 2021.

[52] W. H. Chin, A. G. Constantinides, and D. B. Ward, "Parallel multistage detection for multiple antenna wireless systems," *Electron. Lett.*, vol. 38, pp. 597–599, 2002.

[53] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2017.

[54] G. O. LLC., "Gurobi optimizer reference manual," 2021.

[55] S. L. Loyka, "Channel capacity of MIMO architecture using the exponential correlation matrix," *IEEE Commun. Lett.*, vol. 5, no. 9, pp. 369–371, 2001.

[56] 3GPP, "Study on 3-D channel model for LTE," Tech. Rep. 36.873, 2015.

[57] S. Jaeckel, L. Raschkowski, K. Börner, and L. Thiele, "QuaDRiGa: A 3-D multi-cell channel model with time evolution for enabling virtual field trials," *IEEE Trans. Antennas Propag.*, vol. 62, no. 6, pp. 3242–3256, 2014.

[58] R. Doost-Mohammady, O. Bejarano, L. Zhong, J. R. Cavallaro, E. Knightly, Z. M. Mao, W. W. Li, X. Chen, and A. Sabharwal, "RENEW: Programmable and observable massive MIMO networks," in *Asilomar Conf. on Signals, Syst., and Comp.*, pp. 1654–1658, 2018.

[59] Ö. Özdogan, E. Björnson, and E. G. Larsson, "Massive MIMO with spatially correlated Rician fading channels," *IEEE Trans. Commun.*, vol. 67, no. 5, pp. 3234–3250, 2019.

[60] N. Zilberstein, A. Sabharwal, and S. Segarra, "Solving linear inverse problems using higher-order annealed Langevin diffusion," *arXiv preprint arXiv:2305.05014*, 2023.

[61] K. Venugopal, A. Alkhateeb, N. González Prelcic, and R. W. Heath, "Channel estimation for hybrid architecture-based wideband millimeter wave systems," vol. 35, no. 9, pp. 1996–2009, 2017.

[62] B. N. Bhaskar, G. Tang, and B. Recht, "Atomic norm denoising with applications to line spectral estimation," vol. 61, no. 23, pp. 5987–5999, 2013.

[63] E. Nayebi and B. D. Rao, "Semi-blind channel estimation for multiuser massive MIMO systems," vol. 66, no. 2, pp. 540–553, 2017.

[64] C. Metzler, A. Mousavi, and R. Baraniuk, "Learned D-AMP: Principled neural network based compressive image recovery," vol. 30, 2017.

[65] C.-H. Chao, W.-F. Sun, B.-W. Cheng, Y.-C. Lo, C.-C. Chang, Y.-L. Liu, Y.-L. Chang, C.-P. Chen, and C.-Y. Lee, "Denoising likelihood score matching for conditional score-based data generation," *arXiv preprint arXiv:2203.14206*, 2022.