Generalizing Projected Gradient Descent for Deep-Learning-Aided Massive MIMO Detection

Lanxin He, *Graduate Student Member, IEEE*, Zheng Wang, *Senior Member, IEEE*, Shaoshi Yang, *Senior Member, IEEE*, Tao Liu, and Yongming Huang, *Senior Member, IEEE*

***Abstract*— In this paper, the projected-gradient-descent (PGD) -based detector for massive MIMO system, which consists of two basic operations — projection and gradient descent (GD), is studied to achieve the performance improvement. Since the projection and GD step have different loss functions, necessary compromise has to be made to balance them during iterations. For this reason, the generalized PGD (GPGD) method is proposed with flexible choices of projection and GD. Different from performing projection and GD alternatively, we show that implementing projection after every multiple GD steps is a better solution. Meanwhile, the step-size of GD is also investigated for convergence efficiency. After that, by unfolding this proposed GPGD method with deep neural networks (DNN), the self-corrected auto-detector (SAD) is established to achieve better decoding performance, where enhancement by attention mechanism and extension by another iterative method are also given for performance improvement and efficiency.**

***Index Terms*— Massive MIMO detection, projected gradient descent, denoising auto-encoder, attention mechanism.**

I. Introduction

NOWADAYS, massive multiple-input multiple-output (MIMO) technology acts as an enabler and facilitator for the development of future 6G communication systems owing to its advantages in spectrum and energy efficiency, robustness as well as link reliability [1], [2], [3]. Besides, the uplink signal detection problem plays an important role in the massive MIMO systems, and there exist various traditional detectors to solve this problem [4], [5], [6], [7]. Recently, the deep learning (DL) technology, which has brought unprecedented boosting in various fields [8], [9], [10], [11], has been introduced to massive MIMO detection and drawn the increasing attention. Technically, these DL-based MIMO detection networks can be divided into two types, namely, the data-driven and the model-driven networks.

Requiring a large amount of data to train, the data-driven networks normally involve standard neural networks like feed-forward network. A representative instance of these data-driven networks is DetNet [12], [13], which is established by unfolding the projected gradient descent (PGD) method and capable of achieving better performance than minimum mean square error (MMSE) detector. Besides, in [14] and [15], ScNet and FS-Net introduce sparse connectivity by substituting the Hadamard product for the matrix multiplication in DetNet to alleviate the complexity. BD-Net in [16] employs distinct bidirectional long short-term memory (LSTM) units in DetNet to enhance the performance. WeSNet in [17] adopts profile weight coefficients onto DetNet, resulting in a more complexity-scalable network. Except these works to do with DetNet, LISA in [18] utilizes LSTM units to mimic the procedure of successive interference cancellation (SIC) and achieves the near-optimal performance. Reference [19] combines a feed-forward network into sphere decoding (SD) algorithm to predict the best searching radius. RE-MIMO in [20] is built around recurrent inference machine (RIM) in solving inverse problems and adopts self-attention mechanism in its encoder modular. MRIDA-Net [21] implements the modified randomized iterative detection algorithm (MRIDA) by deep neural networks for a better detection performance.

As for the model-driven networks relying on certain detection models that are relatively mature, a general treatment is to make some specific parameters in the corresponding detection models trainable. For example, [22] establishes the LcgNet by setting the two step-sizes in conjugate gradient descent (CGD) method as trainable, so that the original exact computation is eliminated. OAMP-Net [23] incorporates DL into the orthogonal approximate message passing (OAMP) algorithm by adding trainable scalar variables. Both [24] and [25] involve learning the damping factors in massage passing algorithm. Furthermore, [26] develops a trainable optimizer with respect

Manuscript received 9 October 2022; revised 12 March 2023; accepted 4 June 2023. Date of publication 12 July 2023; date of current version 12 March 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 61801216, Grant 62225107, and Grant 61720106003; in part by ZTE Corporation through Research Program under Grant 2023ZTE01-04; in part by the Natural Science Foundation of Jiangsu Province under Grant BK20180420; in part by the Beijing Municipal Natural Science Foundation under Grant L202012 and Grant Z220004; and in part by the Beijing Municipal Science and Technology Commission under Grant Z221100007722036. An earlier version of this paper was presented in part at the IEEE 94th Vehicular Technology Conference (VTC-Fall), September, 2021 [DOI: 10.1109/VTC2021-Fall52928.2021.9625500]. The associate editor coordinating the review of this article and approving it for publication was C. Huang. *(Corresponding author: Zheng Wang.)*

Lanxin He and Tao Liu are with the College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210000, China (e-mail: lanxin_he@nuaa.edu.cn).

Zheng Wang and Yongming Huang are with the School of Information Science and Engineering, Southeast University, Nanjing 210096, China (e-mail: z.wang@ieee.org).

Shaoshi Yang is with the School of Information and Communication Engineering, Beijing University of Posts and Telecommunications (BUPT), Beijing 100876, China, and also with the Key Laboratory of Universal Wireless Communications, Ministry of Education, Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: shaoshi.yang@bupt.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TWC.2023.3292124.

Digital Object Identifier 10.1109/TWC.2023.3292124

to the updating of damping factors in expectation propagation algorithm, providing a new perspective on the designs of model-driven network by this hyper-network way.

Ever since the PGD method has been successfully unfolded by DetNet [12], numerous works have discussed and utilized it to construct the detection networks. The RIM in [20] comes from the gradient-based inference method for solving the inverse problems, which also provides a perspective to view the PGD-based detector from an inference problem. Reference [27] derives PGD method from the alternating direction method of multiplier (ADMM) in optimization problem. However, the relationship between the projection and gradient descent (GD) step in it has not been clearly studied. In regard to this problem, we carry out a detailed investigation in this paper. The contributions are outlined as follows:

- The relationship of the projection and GD step is revealed from the perspective of their corresponding loss functions. We develop that these two operations iterate in PGD method, until a balance between the two related loss functions is achieved. To the authors' knowledge, this is the first time that the PGD-based detection is viewed through the behaviors of projection and GD step.
- To fully exploit the potential of the above intrinsic property, we generalize the PGD method to a more overall framework, namely the generalized PGD (GPGD) method, for performance improvement. With respect to this GPGD method, the selection about the number of GD steps is also studied in full detail for better decoding performance. Meanwhile, we develop the optimal choice of the step-size to improve the convergence efficiency.
- For the realization of GPGD method, the self-correction auto-detector (SAD) is established by DNN, where two modulars are involved. One is a well-trained denoising auto-encoder (DAE) to implement the projection, while the other is a self-correction modular executing GD step. Then, two enhancement schemes, namely the attention mechanism and the extension of GD steps, are explored for further performance and efficiency improvement.

The rest of this paper is organized as follows. Section II gives a brief introduction to uplink massive MIMO detection problem and PGD-based detection. Besides, the architecture of DAE is also introduced. In Section III, the relationship of the two operations in PGD method, the projection and GD step, is investigated. Based on this, we generalize the original PGD method to a more overall framework, GPGD method. Section IV further analyzes the proposed GPGD method and discusses the choice of its parameters. By unfolding this GPGD method, SAD is established in Section V, where two enhancement schemes are also given, including combination with attention mechanism and extension in regard to the self-correction modular. After that, simulations of the proposed SAD for uplink massive MIMO detection are presented in Section VI, and Section VII concludes the paper.

Notation: Matrices and column vectors are denoted by upper and lowercase boldface letters, and the transpose, inverse, Moore-Penrose inverse of a matrix $\mathbf{B}$ by $\mathbf{B}^T$, $\mathbf{B}^{-1}$, and $\mathbf{B}^\dagger$, respectively. $\Re(\cdot)$ and $\Im(\cdot)$ indicate the real and imaginary components. $\mathbf{I}$ denotes an identity matrix and $\ln(\cdot)$ denotes the natural logarithm. $\mathcal{N}(x;\mu,\sigma) \triangleq \frac{1}{\sqrt{2\pi}\sigma}\exp(-\frac{(x-\mu)^2}{2\sigma^2})$ denotes a Gaussian distribution and $\mathcal{U}(a,b)$ represents a uniform distribution over the interval $[a,b]$. $\lfloor \mathbf{x} \rceil$ gives the nearest integer to $\mathbf{x}$ inside the feasible region.

II. Preliminaries

In this section, the massive MIMO system model and some preliminaries about the PGD-based detection as well as the denoising auto-encoder are introduced.

A. System Model

Considering the standard massive linear MIMO system with N_t transmit and N_r receive antennas, let $\ddot{\mathbf{x}} \in \mathcal{X}^{N_t}$ denote the complex-valued transmitted signal vector, and the corresponding received signal vector $\ddot{\mathbf{y}} \in \mathbb{C}^{N_r}$ is given by

$$\ddot{\mathbf{y}} = \ddot{\mathbf{H}}\ddot{\mathbf{x}} + \ddot{\mathbf{n}}. \tag{1}$$

Here, $\mathcal{X}^{N_t} \subset \{\mathbb{Z}^{N_t} + j\mathbb{Z}^{N_t}\}$ is the set for all possible transmitted M-QAM symbols, $\ddot{\mathbf{H}} \in \mathbb{C}^{N_r \times N_t}$ is the channel matrix, $\ddot{\mathbf{n}} \in \mathbb{C}^{N_r}$ is the additive white Gaussian noise (AWGN) vector with zero mean and variance σ_n^2. This accounts for a real-valued $2N_r \times 2N_t$ system

$$\begin{bmatrix} \Re(\ddot{\mathbf{y}}) \\ \Im(\ddot{\mathbf{y}}) \end{bmatrix} = \begin{bmatrix} \Re(\ddot{\mathbf{H}}) & -\Im(\ddot{\mathbf{H}}) \\ \Im(\ddot{\mathbf{H}}) & \Re(\ddot{\mathbf{H}}) \end{bmatrix} \begin{bmatrix} \Re(\ddot{\mathbf{x}}) \\ \Im(\ddot{\mathbf{x}}) \end{bmatrix} + \begin{bmatrix} \Re(\ddot{\mathbf{n}}) \\ \Im(\ddot{\mathbf{n}}) \end{bmatrix}, \tag{2}$$

which can be further expressed as

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}. \tag{3}$$

For notation simplicity, from now on N and K are utilized to denote $2N_r$ and $2N_t$, respectively. Then, channel matrix $\mathbf{H} \in \mathbb{R}^{N \times K}$ is considered and the problem of massive MIMO detection becomes

$$\widehat{\mathbf{x}} = \arg\min_{\mathbf{x} \in \mathcal{A}^K} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2, \tag{4}$$

where $\mathcal{A} = \{\pm 1, \pm 3, \ldots, \pm\sqrt{M} - 1\}$ with $\sqrt{M}$ representing the modulation index of the corresponding real-valued amplitude-shift keying (ASK).

B. Massive MIMO Detection Based on PGD Method

The detection in (4) can be equivalently reformulated as an optimization problem

$$\min_{\mathbf{x} \in \mathbb{R}^K} \iota_{\mathcal{A}^K}(\mathbf{x}) + g(\mathbf{x}). \tag{5}$$

Here $\iota_{\mathcal{A}^K}$ represents an indicator function of the nonempty set $\mathcal{A}^K \subset \mathbb{R}^K$

$$\iota_{\mathcal{A}^K}(\mathbf{x}) : \mathbf{x} \to \begin{cases} 0, & \text{if } \mathbf{x} \in \mathcal{A}^K; \\ +\infty, & \text{if } \mathbf{x} \notin \mathcal{A}^K, \end{cases}$$

and $g(\mathbf{x})$ is a smooth convex loss function

$$g(\mathbf{x}) = \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2. \tag{6}$$

In order to solve (5), there are many existing linear methods focusing on $g(\mathbf{x})$, such as zero forcing (ZF) and MMSE detectors

$$\widetilde{\mathbf{x}}_{\mathrm{ZF}} = \mathbf{H}^{\dagger}\mathbf{y}, \quad \widetilde{\mathbf{x}}_{\mathrm{MMSE}} = \underline{\mathbf{H}}^{\dagger}\underline{\mathbf{y}}, \tag{7}$$

where $\underline{\mathbf{H}}$ and $\underline{\mathbf{y}}$ are the augmented channel matrix and received signal

$$\underline{\mathbf{H}} = \begin{bmatrix} \mathbf{H} \\ \sigma_{\mathbf{n}}\mathbf{I}_K \end{bmatrix}, \quad \underline{\mathbf{y}} = \begin{bmatrix} \mathbf{y} \\ \mathbf{0}_K \end{bmatrix}. \tag{8}$$

Then, by a simple rounding to constellation $\mathcal{A}^K$, an estimate that lies inside $\mathcal{A}^K$ would be the final result

$$\widehat{\mathbf{x}}_{\mathrm{ZF}} = \lfloor \widetilde{\mathbf{x}}_{\mathrm{ZF}} \rceil \in \mathcal{A}^K, \quad \widehat{\mathbf{x}}_{\mathrm{MMSE}} = \lfloor \widetilde{\mathbf{x}}_{\mathrm{MMSE}} \rceil \in \mathcal{A}^K, \tag{9}$$

so that the constraint related to the term $\iota_{\mathcal{A}^K}(\mathbf{x})$ is satisfied. For lower complexity, various iterative methods that avoid inverse operation are adopted to approximate ZF or MMSE detector. For instance, the GD method updates according to the current gradient $\nabla g(\widetilde{\mathbf{x}}_n)$

$$\widetilde{\mathbf{x}}_{n+1} = \widetilde{\mathbf{x}}_n - \delta_n \nabla g(\widetilde{\mathbf{x}}_n), \quad n = 0, 1, \ldots, L-1, \tag{10}$$

where n is the iteration index, δ_n is the step-size, L is the total iterations. Also a rounding operator is required here to satisfy the indicator function $\iota_{\mathcal{A}^K}$

$$\widehat{\mathbf{x}}_{\mathrm{GD}} = \lfloor \widetilde{\mathbf{x}}_L \rceil. \tag{11}$$

Similarly, the projected gradient descent (PGD) method [28], which combines the projection operator and GD step, can be utilized as an optimization algorithm superior to GD method. By defining a loss function measuring the distance to the transmitted signal $\mathbf{x}_t$

$$f_{\mathbf{x}_t}(\mathbf{x}) = \|\mathbf{x}_{\mathrm{t}} - \mathbf{x}\|^2, \tag{12}$$

the projection $\mathcal{P}$ that minimizes it:

$$\arg\min_{\mathcal{P}} f_{\mathbf{x}_t}(\mathcal{P}(\mathbf{x})), \tag{13}$$

is considered as a well-chosen projection. It can be seen that the transmitted signal $\mathbf{x}_t$ is involved here to do with the decision about this projection. This would introduce the prior information about the transmitted signal at the very beginning of detection, providing the possibilities for performance improvement. Owing to this, the PGD method is likely to outperform the pure GD method, and thus the MMSE detector as well. After a satisfying projection $\mathcal{P}$ has been found, following the traditional setup, the PGD-based detector iterates this projection and GD step alternatively to update the detection estimate

$$\widetilde{\mathbf{x}}_{n+1} = \mathcal{P}(\widetilde{\mathbf{x}}_n - \delta_n \nabla g(\widetilde{\mathbf{x}}_n)) \in \mathbb{R}^K, \ n = 0, 1, \ldots, L-1. \tag{14}$$

Finally, $\widetilde{\mathbf{x}}_L$ is rounded to the constellation $\mathcal{A}^K$ to obtain the final estimation

$$\widehat{\mathbf{x}} = \lfloor \widetilde{\mathbf{x}}_L \rceil \in \mathcal{A}^K. \tag{15}$$

However, finding such an optimal projection is difficult, and a poor-quality projection that does not contribute to detection would reduce PGD method to a pure GD method.

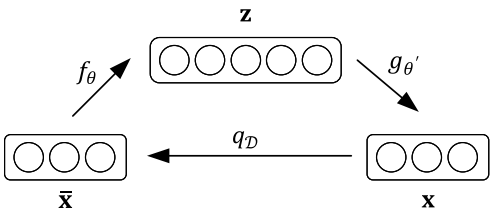

Fig. 1. Architecture of the denoising auto-encoder.

Fortunately, [13] manages to establish a projection operator with deep neural networks (DNN), where by invoking back-propagation (BP) algorithm, the projection can be gradually refined to satisfy the criterion in (13).

C. Denoising Auto-Encoder

After receiving a corrupted data point as input, the *denoising autoencoder* (DAE) is designed to predict the original, uncorrupted one [29]. In Fig.1 we visualize the structure of a DAE, where $q_{\mathcal{D}}$ represents a disturbance procedure performing on $\mathbf{x}$ and produces a corrupted point $\bar{\mathbf{x}}$. For instance, a Gaussian corruption with noise variance σ^2 has the following form

$$q_{\mathcal{D}}(\bar{\mathbf{x}}|\mathbf{x}) \sim \mathcal{N}(\bar{\mathbf{x}}; \mathbf{x}, \sigma^2\mathbf{I}). \tag{16}$$

This auto-encoder provides a scheme to recover the signal $\mathbf{x}$ from a disturbed one. Specifically, with respect to the corrupted $\bar{\mathbf{x}}$, an encoder f_θ, characterized by the learned parameter θ, first works on it and outputs a hidden vector $\mathbf{z}$

$$\mathbf{z} = f_\theta(\bar{\mathbf{x}}). \tag{17}$$

Then, a decoder $g_{\theta'}$ acts on this hidden vector

$$\mathbf{x} = g_{\theta'}(\mathbf{z}), \tag{18}$$

where θ' is another set of trainable parameter and a reconstruction about the original signal $\mathbf{x}$ has been accomplished.

III. Generalized PGD Method

The success of PGD-based detection depends not only on the projection in (13), but also on the GD step as well. Therefore, we reveal the role of these two operations by dividing the process of PGD method into two stages:

- Stage I Before the PGD method reaches the performance of a ZF detector, both the projection and the GD step can benefit each other towards a ZF performance.
- Stage II After the PGD method outperforms ZF detector, the projection continues finding solutions closer to ML detection, while the GD step attracts solutions to a ZF point.

There are two loss functions involved in PGD method: $f_{\mathbf{x}_t}(\mathbf{x})$ and $g(\mathbf{x})$ to do with projection and GD step, respectively. We can show their relationship by noticing

$$\begin{aligned} \sqrt{g(\mathbf{x})} &= \|\mathbf{H}\mathbf{x}_{\mathrm{t}} + \mathbf{n} - \mathbf{H}\mathbf{x}\| \\ &= \|\mathbf{H}(\mathbf{x}_{\mathrm{t}} - \mathbf{x}) + \mathbf{n}\| \\ &\le \|\mathbf{H}(\mathbf{x}_{\mathrm{t}} - \mathbf{x})\| + \|\mathbf{n}\| \\ &\le \|\mathbf{H}\| f_{\mathbf{x}_t}(\mathbf{x}) + \|\mathbf{n}\|, \end{aligned} \tag{19}$$

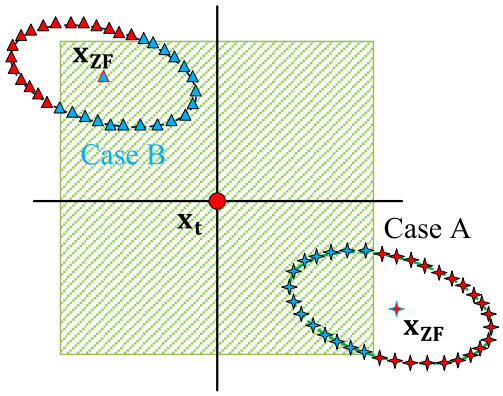

Fig. 2. Illustration of the two cases. Case A: $\mathbf{x}_{\mathrm{ZF}}$ lies outside the success region; Case B: $\mathbf{x}_{\mathrm{ZF}}$ lies inside the success region.

where $\|\mathbf{H}\|$ can be any arbitrary matrix norm of $\mathbf{H}$. Hence the decrease of $f_{\mathbf{x}_t}(\mathbf{x})$ would lower an upper bound for $g(\mathbf{x})$, encouraging it to decrease, which explains why the projection can accelerate the convergence of GD step at stage I. On the other hand, since GD step acts as a promising iterative method leading to a ZF solution, its positive influence on the projection at stage I is obvious. However at stage II, the minimum of $g(\widetilde{\mathbf{x}})$

$$\min_{\mathbf{x}\in\mathbb{R}^K} g(\widetilde{\mathbf{x}}) = g(\widetilde{\mathbf{x}}_{\mathrm{ZF}}) \stackrel{\triangle}{=} g^* \tag{20}$$

serves as an absolute lower bound for $g(\mathbf{x})$. This minimum value g^* corresponds to the limitation that GD steps can achieve. Also considering that reaching the minimum of $g(\mathbf{x})$ is not equivalent to finding the minimum of $f_{\mathbf{x}_t}(\mathbf{x})$, an adversarial behavior between these two processes should be expected once this minimum g^* has been reached, which can be formulated as

$$f_{\mathbf{x}_t}(\mathcal{P}(\widetilde{\mathbf{x}})) < f_{\mathbf{x}_t}(\widetilde{\mathbf{x}}) < f_{\mathbf{x}_t}(\mathrm{GD}[\widetilde{\mathbf{x}}]), \tag{21a}$$

$$g(\mathcal{P}(\widetilde{\mathbf{x}})) > g(\widetilde{\mathbf{x}}) > g(\mathrm{GD}[\widetilde{\mathbf{x}}]). \tag{21b}$$

Here $\mathrm{GD}[\cdot]$ denotes a single GD step as (10). This implies that at stage II, a GD step decreases $g(\widetilde{\mathbf{x}})$ while it would increase $f_{\mathbf{x}_t}(\widetilde{\mathbf{x}})$, and the situation is opposite for a projection.

At this point, a natural solution to this adversarial behavior should be just eliminating the GD step at stage II if it keeps pulling the points backward to the ZF point. However, we point out that the pure projection at stage II is not an ideal answer, because the projection itself introduces mistakes into the system. In other words, only decreasing $f_{\mathbf{x}_t}(\widetilde{\mathbf{x}})$ is not necessarily equivalent to the improvement of detection performance. This can also be seen from the later numerical results in Fig.8, where after each projection, even though the $f_{\mathbf{x}_t}(\widetilde{\mathbf{x}})$ value does decrease, the detection performance deteriorates. To formulate this statement, we assume that a detector $\mathcal{D}_1$ is more powerful in reducing $f_{\mathbf{x}_t}$ value than another one $\mathcal{D}_2$ and use $g(\lfloor\widetilde{\mathbf{x}}\rceil)$ to estimate the detection performance since the ML criterion is considered.[1]

$$\Pr(g(\lfloor\widetilde{\mathbf{x}}_1\rceil) > g(\lfloor\widetilde{\mathbf{x}}_2\rceil) | f_{\mathbf{x}_t}(\widetilde{\mathbf{x}}_1) < f_{\mathbf{x}_t}(\widetilde{\mathbf{x}}_2)) > 0, \tag{22}$$

where $\widetilde{\mathbf{x}}_1 = \mathcal{D}_1(\widetilde{\mathbf{x}})$ and $\widetilde{\mathbf{x}}_2 = \mathcal{D}_2(\widetilde{\mathbf{x}})$ are estimates of the two detectors before the final rounding to constellation. To explain this limitation of projection, with each particular $\mathbf{x}_t$, we define its corresponding *success region*, depicted by the green dash-lined square in Fig.2. Once an estimated solution $\widetilde{\mathbf{x}}$ is located inside this region, rounding it to the constellation definitely gives the transmitted signal and leads to a successful detection. If a point $\widetilde{\mathbf{x}}$ is already a correct result (inside the success region), then further decrease in $f_{\mathbf{x}_t}(\widetilde{\mathbf{x}})$ would attract $\widetilde{\mathbf{x}}$ to $\mathbf{x}_t$ further, while for this particular point, being placed nearer to $\mathbf{x}_t$ would not contribute to any performance gain since it is already able to be detected correctly. However, as the adversarial property of the two loss functions implies, decrease in $f_{\mathbf{x}_t}(\widetilde{\mathbf{x}})$ results in the increase of $g(\widetilde{\mathbf{x}})$, making it possible for detection to deteriorate.

These possible mistakes introduced by projection can be further specified as illustrated in Fig.2, where those points located on an ellipse[2] enjoy the same value of $g(\widetilde{\mathbf{x}})$. Let $\triangle g(\widetilde{\mathbf{x}}) = g(\widetilde{\mathbf{x}}) - g^*$ measures the difference between $g(\widetilde{\mathbf{x}})$ value of these points and the minimum g^*. When $\widetilde{\mathbf{x}}_{\mathrm{ZF}}$ locates outside the success region (Case A, depicted as a star), a nonzero $\triangle g(\widetilde{\mathbf{x}})$ makes it possible for $\widetilde{\mathbf{x}}$ to relocate inside the success region. This means that for an estimate given by PGD method $\widetilde{\mathbf{x}}_{\mathrm{PGD}}$, once $\triangle g(\widetilde{\mathbf{x}}_{\mathrm{PGD}}) > \epsilon_{\mathrm{A}}$, where ϵ_{A} is a small-valued threshold, a *positive probability* defined by

$$P^{\mathrm{positive}} = \Pr(\lfloor\widetilde{\mathbf{x}}_{\mathrm{PGD}}\rceil = \mathbf{x}_t | \lfloor\widetilde{\mathbf{x}}_{\mathrm{ZF}}\rceil \neq \mathbf{x}_t) \tag{23}$$

would not be zero anymore, implying the potential that a received signal $\mathbf{y}$ would get detected correctly even it could have been mis-detected by the pure GD method or the MMSE detector. This is where the possible performance gain of PGD method compared to the pure GD comes from. The other case is when $\widetilde{\mathbf{x}}_{\mathrm{ZF}}$ already locates inside the success region (Case B, depicted as a triangular). Clearly, $\triangle g(\widetilde{\mathbf{x}}_{\mathrm{PGD}}) > \epsilon_{\mathrm{B}}$ results in a non-zero *negative probability*

$$P^{\mathrm{negative}} = \Pr(\lfloor\widetilde{\mathbf{x}}_{\mathrm{PGD}}\rceil \neq \mathbf{x}_t | \lfloor\widetilde{\mathbf{x}}_{\mathrm{ZF}}\rceil = \mathbf{x}_t), \tag{24}$$

where ϵ_B is another certain disturbance. In this case, a received signal $\mathbf{y}$ that could have been correctly detected by simple GD iterations is mistaken by PGD, which is signified as the aforementioned mistakes introduced by projection. What is more, these mistakes made by a projection may still be a burden for a single GD step to correct.

Remark 1: Both projection and GD step are indispensable in PGD-based detection, and it is unnecessary to implement only a single GD step right after each projection.

Considering that the essential change behind these two operations is the evolution of the two loss functions, we generalize the PGD method to a GPGD one, as shown in Fig.3. Specifically, GD would be performed up to m times before the upcoming projection and this iteration is repeated up to P

[1]For an estimate $\widetilde{\mathbf{x}} \in \mathbb{R}^K$, from the ML criterion in (4), after rounding it to the constraint set $\mathcal{A}^K$, a smaller $g(\lfloor\widetilde{\mathbf{x}}\rceil)$ value corresponds to a final solution $\lfloor\widetilde{\mathbf{x}}\rceil$ that has higher possibility in turning out to be a ML solution. Then there is still possibility that $\mathcal{D}_2$ outperforms $\mathcal{D}_1$ in terms of detection.

[2]The points on the ellipse indicate those enjoying the same value of $g(\widetilde{\mathbf{x}}) = ||\mathbf{y} - \mathbf{H}\widetilde{\mathbf{x}}||^2$. In the lattice region, the candidates points $\mathbf{H}\widetilde{\mathbf{x}}$' s should lie on a circle centered at $\mathbf{y}$ if $\widetilde{\mathbf{x}}$' s have equal $g(\widetilde{\mathbf{x}})$. Then, by applying the inverse transformation of $\mathbf{H}$ on the coordinate, this circle would be stretched to an ellipse.

Algorithm 1 Proposed GPGD Method

Input $\mathbf{H}$, $\mathbf{y}$, m, P, δ

Output $\widehat{\mathbf{x}}$: estimated transmit signal

1: **Initialize:** set $\widetilde{\mathbf{x}}_j^{<n>}$ as a random point in $\mathcal{A}^K$; $j=0$; $n=0$.
2: **for** $n=0,1,\ldots,P$ **do**
3: **for** $j=0,1,\ldots,m-1$ **do**
4: $\widetilde{\mathbf{x}}_{j+1}^{<n>}=\mathrm{GD}_\delta[\widetilde{\mathbf{x}}_j^{<n>}]$
5: **end for**
6: $\widetilde{\mathbf{x}}_0^{<n+1>}=\mathcal{P}(\widetilde{\mathbf{x}}_m^{<n>})$
7: **end for**
8: $\widehat{\mathbf{x}}=\lfloor\widetilde{\mathbf{x}}_0^{<P>}\rceil$

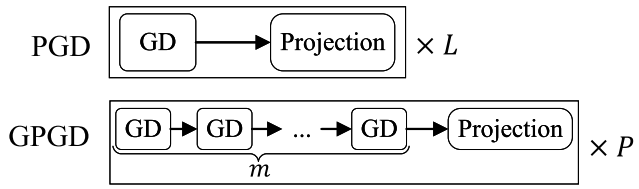

Fig. 3. PGD (upper) and proposed GPGD method (bottom).

times. This updated scheme can be expressed as follows

$$\widetilde{\mathbf{x}}_{j+1}^{<n>}=\mathrm{GD}_\delta[\widetilde{\mathbf{x}}_j^{<n>}],\quad j=0,1,\ldots,m-1; \tag{25a}$$

$$\widetilde{\mathbf{x}}_0^{<n+1>}=\mathcal{P}(\widetilde{\mathbf{x}}_m^{<n>}),\quad n=0,1,\ldots,P-1, \tag{25b}$$

where $\mathrm{GD}_\delta[\cdot]$ specifies a GD step with step-size δ. $\widetilde{\mathbf{x}}_j^{<n>}$ represents the point output by the j-th GD step that follows the n-th projection. To summarize, this proposed GPGD method is outlined in Algorithm 1. With respect to the original PGD framework, one iteration is composed of a GD step and a projection, while in the proposed GPGD framework, the one-time GD step has been extended to a m-times one. For one single iteration, the complexity has indeed been increased due to this modification. However, notice that in most cases, a projection operation is far more time-consuming than a GD step. In other words, the complexity of the PGD method is dominated by the projection operation. At the same time, as would be shown in simulations, the total number of the projections is significantly reduced by this proposed GPGD framework. Therefore, beside the advantages in detection efficiency that the GPGD method brings, the optimization in terms of complexity can also be seen.

IV. Performance Analysis and Optimization

In the proposed GPGD, the steps m and step-size δ relate to the efficiency and convergence of the detection. With respect to these two predefined parameters, we examine their roles by further analysis and develop the suitable choices for them.

A. The Choice of m

As discussed in Sec.III, an ideal relaxation on $g(\widetilde{\mathbf{x}})$ lies between ϵ_A and ϵ_B,

$$\epsilon_\mathrm{A}<\triangle g(\widetilde{\mathbf{x}})\le\epsilon_\mathrm{B}. \tag{26}$$

In fact, with respect to this upper bound ϵ_B, a lower bound on m can be derived. To fully exploit the characteristics of massive MIMO channels, we introduce the quadratic form of $g(\mathbf{x})$ during the upcoming analysis:

$$h(\mathbf{x})=\frac{1}{2}\mathbf{x}^\mathrm{T}\mathbf{A}\mathbf{x}-\mathbf{x}^\mathrm{T}\mathbf{b}, \tag{27}$$

where $\mathbf{A}=\mathbf{H}^\mathrm{T}\mathbf{H}$ is positive definite [30], $\mathbf{b}=\mathbf{H}^\mathrm{T}\mathbf{y}$. A simple calculation shows that $g(\mathbf{x})$ and $h(\mathbf{x})$ have the same first-order and second-order derivatives with respect to $\mathbf{x}$, indicating the equivalence between these two objective functions in terms of updating equation and global minimizer $\mathbf{x}^*$. Then, the GD step can be rewritten in terms of $h(\mathbf{x})$

$$\mathbf{x}_{j+1}=\mathbf{x}_j-\delta\nabla h(\mathbf{x}_j). \tag{28}$$

By defining a weighted norm [31] of $\mathbf{x}$ with respect to the definite positive matrix $\mathbf{A}$

$$\|\mathbf{x}\|_\mathbf{A}^2=\mathbf{x}^\mathrm{T}\mathbf{A}\mathbf{x}, \tag{29}$$

it can be shown that

$$\begin{aligned}&\|\mathbf{x}_j-\mathbf{x}^*\|_\mathbf{A}^2-\|\mathbf{x}_{j+1}-\mathbf{x}^*\|_\mathbf{A}^2\\&\overset{(a)}{=}2\delta\nabla h(\mathbf{x}_j)^T\mathbf{A}(\mathbf{x}_j-\mathbf{x}^*)-\delta^2\nabla h(\mathbf{x}_j)^T\mathbf{A}\nabla h(\mathbf{x}_j).\end{aligned} \tag{30}$$

Here (a) holds due to (28), and also notice that

$$\begin{aligned}\|\mathbf{x}_j-\mathbf{x}^*\|_\mathbf{A}^2&=(\mathbf{x}_j-\mathbf{x}^*)^T\mathbf{A}(\mathbf{x}_j-\mathbf{x}^*)\\&=(\mathbf{x}_j-\mathbf{x}^*)^T\mathbf{A}^T\mathbf{A}^{-1}\mathbf{A}(\mathbf{x}_j-\mathbf{x}^*)\\&\overset{(b)}{=}\nabla h(\mathbf{x}_j)^T\mathbf{A}^{-1}\nabla h(\mathbf{x}_j),\end{aligned} \tag{31}$$

where (b) comes from

$$\mathbf{A}(\mathbf{x}_j-\mathbf{x}^*)=\mathbf{A}\mathbf{x}_j-\mathbf{b}=\nabla h(\mathbf{x}_j). \tag{32}$$

Combining (30) and (31), we get

$$\|\mathbf{x}_{j+1}-\mathbf{x}^*\|_\mathbf{A}^2=\left(1-\frac{2\delta\nabla_j^T\nabla_j+\delta^2\nabla_j^T\mathbf{A}\nabla_j}{\nabla_j^T\mathbf{A}^{-1}\nabla_j}\right)\|\mathbf{x}_j-\mathbf{x}^*\|_\mathbf{A}^2, \tag{33}$$

where we use (32) again and denote $\nabla_j=\nabla h(\mathbf{x}_j)$ for simplicity.

Lemma 1: For a positive definite symmetric $K\times K$ matrix $\mathbf{A}$, there holds at every step j

$$\|\mathbf{x}_{j+1}-\mathbf{x}^*\|_\mathbf{A}^2\le[2-(\Lambda+1)^2]\|\mathbf{x}_j-\mathbf{x}^*\|_\mathbf{A}^2, \tag{34}$$

where $\Lambda=\delta\lambda_1$, and λ_1 is the smallest eigenvalue of $\mathbf{A}$.

Proof: Let the eigenvalues $\lambda_1,\lambda_2,\ldots\lambda_n$ of $\mathbf{A}$ satisfy

$$0<\lambda_1\le\lambda_2\ldots\le\lambda_n. \tag{35}$$

With respect to the term inside the bracket of (33), the following equation holds for any arbitrary non-zero vector $\mathbf{s}$:

$$\begin{aligned}\frac{2\delta\mathbf{s}^T\mathbf{s}+\delta^2\mathbf{s}^T\mathbf{A}\mathbf{s}}{\mathbf{s}^T\mathbf{A}^{-1}\mathbf{s}}&=\frac{2\delta\sum_{i=1}^K\mathbf{s}_i^2+\delta^2\sum_{i=1}^K\lambda_i\mathbf{s}_i^2}{\sum_{i=1}^K(\mathbf{s}_i^2/\lambda_i)}\\&=\frac{2\delta+\delta^2(\sum_{i=1}^K\lambda_i\mathbf{s}_i^2/\sum_{i=1}^K\mathbf{s}_i^2)}{(\sum_{i=1}^K\mathbf{s}_i^2/\lambda_i)/\sum_{i=1}^K\mathbf{s}_i^2}\\&\overset{(c)}{=}\frac{2\delta+\delta^2\sum_{i=1}^K\lambda_i\eta_i}{\sum_{i=1}^K\eta_i/\lambda_i}\\&\triangleq\frac{2\delta+\delta^2\phi(\boldsymbol{\eta})}{\psi(\boldsymbol{\eta})},\end{aligned} \tag{36}$$

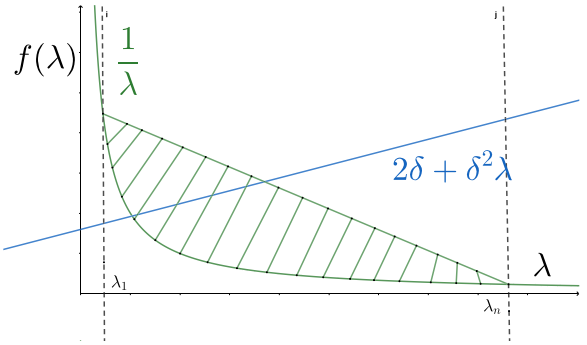

Fig. 4. Values of $\delta^2\lambda + 2\delta$ and $\frac{1}{\lambda}$.

where we denote $\eta_i = \mathbf{s}_i^2 / \sum_{i=1}^K \mathbf{s}_i^2$ at the equality (c). Specifically, $\phi(\boldsymbol{\eta}) = \sum_{i=1}^K \lambda_i \eta_i$ represents a point between λ_1 and λ_n, and thus the numerator characterizes a point at the curve $\delta^2\lambda + 2\delta$. On the other hand, $\psi(\boldsymbol{\eta}) = \sum_{i=1}^K \eta_i/\lambda_i$ is a convex combination of $\frac{1}{\lambda_i}$. Therefore, as depicted in Fig.4, its value is represented by points in the shaded region. Also, with the same $\boldsymbol{\eta}$, λ_i in these two functions are weighted in the same way, corresponding to a certain vertical line in the plot. Clearly the minimum of (36) is reached when it takes the smallest eigenvalue λ_1. Therefore, an appropriate bound for (36) is

$$\frac{2\delta + \delta^2\phi(\boldsymbol{\eta})}{\psi(\boldsymbol{\eta})} \geq \frac{2\delta + \delta^2\lambda_1}{1/\lambda_1} = \Lambda^2 + 2\Lambda. \tag{37}$$

To this end, it is straightforward to get (34). ∎

Theorem 1: A lower bound on m in regard to ϵ_{B} is

$$m \geq \log_{\beta_{\mathrm{GD}}} \frac{r}{\epsilon_{\mathrm{B}}}, \tag{38}$$

where $\beta_{\mathrm{GD}} = \frac{1}{2-(\Lambda+1)^2}$.

Proof: According to Lemma 1, an induction on (34) gives

$$\|\mathbf{x}_m - \mathbf{x}^*\|_{\mathbf{A}}^2 \leq [2 - (\Lambda+1)^2]^m \|\mathbf{x}_0 - \mathbf{x}^*\|_{\mathbf{A}}^2. \tag{39}$$

Considering that the weighted norm measures the difference between the current objective value and the optimal value

$$\frac{1}{2}\|\mathbf{x} - \mathbf{x}^*\|_{\mathbf{A}}^2 = h(\mathbf{x}) - h(\mathbf{x}^*), \tag{40}$$

combining it with the upper bound ϵ_{B} gives

$$h(\mathbf{x}_m) - h(\mathbf{x}^*) \leq [2 - (\Lambda+1)^2]^m r \leq \epsilon_{\mathrm{B}}, \tag{41}$$

where we use a constant $r = h(\mathbf{x}_0) - h(\mathbf{x}^*)$ to represent the residual loss value after the previous projection. Therefore, the number of GD steps m can be lower-bounded by

$$m \geq \frac{\ln \frac{r}{\epsilon}}{\ln \frac{1}{2-(\Lambda+1)^2}} = \log_{\beta_{\mathrm{GD}}} \frac{r}{\epsilon_B}, \tag{42}$$

where $\beta_{\mathrm{GD}} = \frac{1}{2-(\Lambda+1)^2}$ depends on the step-size δ as well as the smallest eigenvalue of $\mathbf{A}$ and would be a number greater than 1, $\beta_{\mathrm{GD}} > 1$, if the step-size δ is chosen according to the condition we propose in Sec.IV-B. ∎

The upper bound on m is difficult to develop based on ϵ_A. However, suppose $\triangle g(\widetilde{\mathbf{x}})$ be extremely small, then the result $\widetilde{\mathbf{x}}$ would be fixed on $\widetilde{\mathbf{x}}_{\mathrm{ZF}}$. Accordingly, its $f_{\mathbf{x}_t}$ value would be limited close to $f_{\mathbf{x}_t}(\widetilde{\mathbf{x}}_{\mathrm{ZF}})$, leading to a constrained $f_{\mathbf{x}_t}(\widetilde{\mathbf{x}})$. Therefore, to assure not performing GD steps too much to maintain the decrease of $f_{\mathbf{x}_t}$ value, we bound this value after m GD steps by the value of its ancestor iteration as in the first inequality of the following *principle of selecting* m

$$f_{\mathbf{x}_t}(\mathbf{x}_m^{<t>}) < f_{\mathbf{x}_t}(\mathbf{x}_m^{<t-1>}), \tag{43a}$$
$$\triangle g(\mathbf{x}_m^{<t>}) \leq \epsilon_{\mathrm{B}}, \tag{43b}$$

where the second inequality is in accordance with Theorem 1. (43a) gives a criterion on the decision of the largest m that could be adopted. This one is easy to fulfill since in practice m would not be chosen to be too large due to the consideration for complexity. While (43b) shows that the lower bound of m depends on the residual loss r and the GD base β_{GD}. On one hand, as the detection goes on, it is safe to use a smaller m since the residual loss would decrease accordingly. Moreover, in the next chapter, some extension of GD step has been discussed and at this point the suitable m would change as the GD base β_{GD} has changed.

B. The Choice of Step-Size δ

GD steps play an important role in GPGD method, while the success of GD depends on the effective choice of δ.

Theorem 2: For a massive MIMO system, to ensure the convergence of GPGD method, the suitable step-size δ lies in the following range:

$$0 < \delta < \min\left(\frac{1}{N_r(1+1/\sqrt{\alpha})^2}, \frac{\sqrt{2}-1}{N_r(1-1/\sqrt{\alpha})^2}\right), \tag{44}$$

where $\alpha = N_r/N_t$ is the antenna ratio.

Proof: In order to develop a reasonable choice of δ, we first explicate the role of it by induction over j in a GD updating step:

$$\begin{aligned}\mathbf{x}_j &= (\mathbf{I} - 2\delta\mathbf{A})\mathbf{x}_{j-1} + 2\delta\mathbf{b} \\ &= (\mathbf{I} - 2\delta\mathbf{A})^j\mathbf{x}_0 + \sum_{i=0}^{j-1}(\mathbf{I} - 2\delta\mathbf{A})^i(2\delta\mathbf{b}).\end{aligned} \tag{45}$$

Now we consider $\mathbf{G} = \mathbf{I} - 2\delta\mathbf{A}$. According to the theory of matrix power series [30], for a $N_t \times N_t$ matrix $\mathbf{G}$, $\mathbf{G}^t$ converges to $\mathbf{0}_{N_t}$ when the spectral radius of $\mathbf{G}$, denoted by $\rho(\mathbf{G})$, is less than 1, which means the following inequality needs to be held:

$$\rho(\mathbf{G}) = \rho(\mathbf{I} - 2\delta\mathbf{A}) < 1. \tag{46}$$

In this case, (45) gives a simple form: $\mathbf{x}_j = \mathbf{A}^{-1}\mathbf{b} = \mathbf{x}_{\mathrm{ZF}}$. Moreover, satisfying (46) is equivalent to bound the absolute value of every eigenvalue of $\mathbf{G}$, $\lambda(\mathbf{G})$, under 1. Then (46) changes to

$$|1 - 2\delta\lambda(\mathbf{A})| < 1 \implies 0 < \delta\lambda(\mathbf{A}) < 1. \tag{47}$$

Since $\mathbf{A}$ is positive definite with positive eigenvalues, (47) requires

$$0 < \delta < \frac{1}{\lambda_n}, \tag{48}$$

where, again, λ_n is the largest eigenvalue of $\mathbf{A}$.

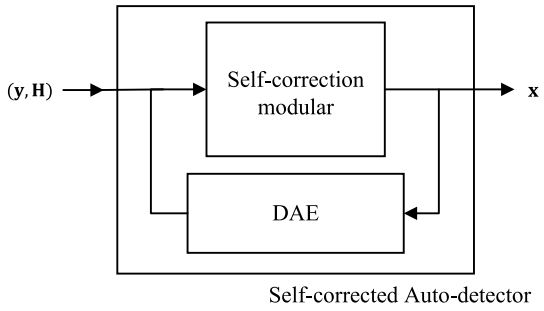

Fig. 5. Structure of the proposed SAD.

Meanwhile, recall that in Lemma 1, the residual of $h(\mathbf{x})$ shrinks obeying (34). Hence, to guarantee the stability of GD steps, it is necessary to satisfy $2-(\Lambda+1)^2>0$, which gives

$$\frac{-\sqrt{2}-1}{\lambda_1}<\delta<\frac{\sqrt{2}-1}{\lambda_1}. \tag{49}$$

Combining this with (48), we obtain

$$0<\delta<\min(\frac{1}{\lambda_n},\frac{\sqrt{2}-1}{\lambda_1}), \tag{50}$$

where a threshold for $\min(\frac{1}{\lambda_n},\frac{\sqrt{2}-1}{\lambda_1})$ occurs when the two elements are equal, giving

$$\kappa=\sqrt{\frac{\lambda_n}{\lambda_1}}=\frac{1}{\sqrt{2}-1}. \tag{51}$$

Here κ is the condition number of the complex Wishart matrix $\mathbf{A}$, and as N_r grows, according to [32], the largest and smallest eigenvalue of $\mathbf{A}$ converge *in probability* to

$$\lambda_1\xrightarrow{p}N_r(1-1/\sqrt{\alpha})^2;\quad \lambda_n\xrightarrow{p}N_r(1+1/\sqrt{\alpha})^2. \tag{52}$$

Therefore, it is straightforward to arrive at (44). ■

It is worth noticing that the threshold in (51) accounts for an antenna ratio $\alpha\approx 22$, which is rarely occurred in practical application. Therefore, bounding δ by $0<\delta<1/\left[N_r(1+1/\sqrt{\alpha})^2\right]$ would be enough for most cases. Furthermore, we point out that in GPGD method, those GD procedures separated by the projection can be treated independently without considering the projection part as we do.

V. Proposed Self-Corrected Auto-Detector

As for the realization of the proposed GPGD method, we draw inspiration from DAE and establish a self-corrected auto-detector (SAD), which consists of a DAE and a self-correction modular. Its complexity is affordable under the context of massive MIMO system. What is more, we manage to augment its DAE by adopting attention mechanism and extend its self-correction modular for a faster realization.

A. Self-Corrected Auto-Detector

The whole architecture of SAD is pictured in Fig.5, where the SAD has been independently divided into two parts: a self-correction modular and a DAE, with the following updating equations at n-th iteration

$$\mathbf{q}_n=\mathrm{GD}_\delta^m[\widetilde{\mathbf{x}}_{n-1}], \tag{53a}$$

$$\mathbf{z}_n=k\text{-winners}(\mathcal{T}_z^S([\mathbf{q}_n,\mathbf{z}_{n-1}])), \tag{53b}$$

$$\widetilde{\mathbf{x}}_n=\mathcal{T}_x^S(\mathbf{z}_n). \tag{53c}$$

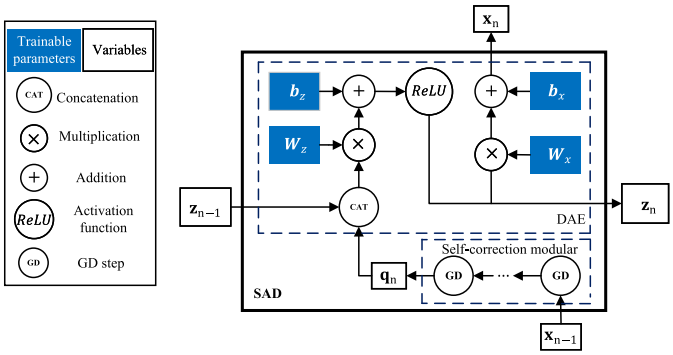

Fig. 6. The inner operations in SAD for one iteration.

Here GD_δ^m in (53a) represents m successive GD steps with a predefined step-size δ, and serves as a self-correction modular to correct the mistakes made by DAE. (53b) and (53c) are treated as the encoder f_θ and decoder $g_{\theta'}$ respectively of a DAE, as in (17) and (18), except that the input of encoder becomes the concatenation to the previous hidden vector $\mathbf{z}_{n-1}$, $[\mathbf{q}_n,\mathbf{z}_{n-1}]$. Meanwhile, the k-winners activation function and the sparse layers, $\mathcal{T}_z^S$ and $\mathcal{T}_x^S$, as in [33] and [34] are adopted here, where the same sparsity S is assumed. By doing so, a sparsity representation proposed in [33] is introduced, which reduces the connections required to train, and thus helps avoiding over-fitting during training.

The total trainable parameters of our proposed SAD are those constituting $\mathcal{T}_z^S:\mathbb{R}^{K+\nu K}\to\mathbb{R}^{\nu K}$ as well as $\mathcal{T}_x^S:\mathbb{R}^{\nu K}\to\mathbb{R}^K$ and denoted as

$$\theta_{\mathrm{SAD}}=\{\mathbf{W}_z,\mathbf{b}_z,\mathbf{W}_x,\mathbf{b}_x\}. \tag{54}$$

It can be noticed that the parameter sharing structure [35] is adopted, which reduces the storage complexity but poses a threat of vanishing gradient [36]. However, since the projection times have been significantly reduced due to the adoption of GPGD method, this risk is under control. Moreover, as the robustness to noise is expected here, this auto-encoder is chosen to be *overcomplete* [29], in which the hidden vector has the dimension greater than the input. This means that the coefficient ν is an integer greater than one, $\nu>1$. The detailed operations in SAD for one iteration is illustrated in Fig.6.

B. Complexity Analysis

In the sequel we present the space complexity, of SAD and its time complexity analysis. The trainable parameters in SAD stem from

$$\begin{aligned}|\theta|&=\underbrace{S\nu K\times(K+\nu K)}_{\mathbf{W}_z}+\underbrace{\nu K}_{\mathbf{b}_z}+\underbrace{S\nu K\times K}_{\mathbf{W}_x}+\underbrace{K}_{\mathbf{b}_x}\\&=\mathcal{O}(S\nu^2K^2).\end{aligned} \tag{55}$$

It conveys that in SAD, owing to the parameter sharing structure, the amount of trainable weights is independent of the total iterations, P, so that the training burden is significantly eased. For comparison, we present the corresponding total parameters required to train in DetNet and SAD in Table I, where the setting for DetNet is in accordance with the 15-layers one in Fig.16. From it, we can see that the trainable parameters of the SAD are much less than those of the DetNet.

TABLE I
SPACE COMPLEXITY COMPARISON

Network	DetNet	SAD ($\nu = 8$)		SAD ($\nu = 12$)	
		S = 0.2	S = 0.3	S = 0.2	S = 0.3
parameters	217950	16672	24864	34822	52026

The time complexity of SAD is as follows. The complexity of the self-correction modular in (53a) is of order $\mathcal{O}(m\alpha K^2)$ for m GD steps, while for the DAE, the complexity of the encoder and decoder in (53b) and (53c) are of order $\mathcal{O}(\nu^2 K^2)$ and $\mathcal{O}(\nu K^2)$, respectively. Then, when $m\alpha < \nu^2$, which is a common setting for most cases, the total complexity of SAD is dominated by that of the encoder in DAE, namely $\mathcal{O}(\nu^2 K^2)$. This is acceptable under the background of massive MIMO, where the influence of hidden-size-coefficient ν is less significant compared to the massive amount of antennas.

C. Enhanced DAE by Attention Mechanism

Improving the ability to reduce $f_{\mathbf{x}_t}$ can help make the detection more efficient. Therefore we include the attention mechanism in SAD and explain the gain from this augmentation. First we rewrite the $f_{\mathbf{x}_t}$ loss to get an equivalent loss f_{equi}. By defining a likelihood value vector of i-th component

$$\mathbf{l}_i = [\exp(-(\widetilde{x}_i - s)^2)]^{\mathrm{T}} \quad s \in \mathcal{A}, \tag{56}$$

its estimated probability of i-th component is given by

$$\mathbf{q}_i = \text{softmax}(\mathbf{l}_i) = \frac{\exp(\mathbf{l}_i)}{\sum_{s\in\mathcal{A}} \exp(l_i(s))}. \tag{57}$$

Then the equivalent loss f_{equi} can be expressed as

$$f_{\text{equi}} = \sum_{i=1}^{2N_t} \log q_i(s = s_t), \tag{58}$$

where $q_i(s = s_t)$ is the estimated probability that the i-th component of the signal $\mathbf{s}_i$ equals to its label s_t. Here (58) is derived from the multi-class cross-entropy loss

$$L_{\text{crossentropy}} = \sum_{i=1}^{2N_t} \sum_{s\in\mathcal{A}} p_i(s) \log q_i(s) \tag{59}$$

where $p_i(s)$ is the label distribution for $\mathbf{s}_i$, which can be represented by a one-hot vector. For instance, in the case of $M = 16$, the label probability of sending $s_i = -3$ is

$$\mathbf{p}_i = [1, 0, 0, 0]^T, \tag{60}$$

or equivalently

$$p_i(s = -3) = 1. \tag{61}$$

Treating $\mathbf{x}_{\text{SAD}}$ as a corruption of $\mathbf{x}_t$, [37] has derived that the mutual information between $\mathbf{X}_t$ and $\mathbf{Z}$, $\mathbf{I}(\mathbf{X}_t; \mathbf{Z})$, has a lower bound and maximizing this lower bound accounts for minimizing the expected reconstruction error

$$\max_{\theta} \mathbf{I}(\mathbf{X}_t; \mathbf{Z}) \geq \max_{\theta,\theta'} \mathbb{E}_{q(\mathbf{X}_t,\mathbf{Z})}[\log p(\mathbf{X}_t|\mathbf{Z})] \tag{62}$$

$$= \min_{\theta,\theta'} \mathbb{E}_{q(\mathbf{X}_t,\mathbf{X}_{\text{SAD}})}[f_{\text{equi}}]. \tag{63}$$

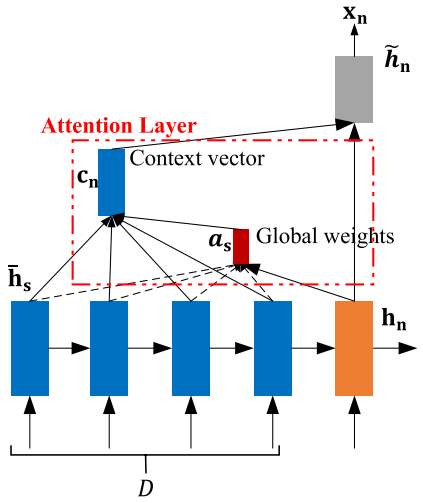

Fig. 7. An illustration of global attention mechanism.

This states the equivalence between the increase of $\mathbf{I}(\mathbf{X}_t; \mathbf{Z})$ and the decrease of $f_{\mathbf{x}_t}(\mathbf{x})$, which implies that any approach boosting the mutual information $\mathbf{I}(\mathbf{X}_t; \mathbf{Z})$ benefits the minimization of the loss function $f_{\mathbf{x}_t}(\mathbf{x})$. Following this intuition, a natural strategy to improve this information maintenance at t-th projection is to collect the information of the previous hidden vectors.

For this purpose, we adopt the attention mechanism, the main idea of which is to take into account all the previous hidden vectors $\bar{\mathbf{h}}_s$, namely source vectors. Then the similarity between the current hidden vector $\mathbf{h}_n$ and those source vectors $\bar{\mathbf{h}}_s$ is measured by a specific scoring function, such as

$$\text{score}(\bar{\mathbf{h}}_s, \mathbf{h}_n) = \mathbf{h}_n^{\mathrm{T}} \cdot \mathbf{W}_a \cdot \bar{\mathbf{h}}_s, \tag{64}$$

where $\mathbf{W}_a$ is the attention weight matrix that needs to quantify during training. Based on the score, the attention coefficients with respect to these source vectors $\bar{\mathbf{h}}_s$ are computed

$$a_s = \text{softmax}(\text{score}(\bar{\mathbf{h}}_s, \mathbf{h}_n)) \tag{65}$$

and determine the weights of each source vector when calculating the context vector

$$\mathbf{c}_n = \sum_{s=1}^{D} a_s \cdot \bar{\mathbf{h}}_s. \tag{66}$$

Here D is the window length and in this global case it equals to the total number of the existed source vectors and changes at each iteration. To this end, together with the current hidden vector $\mathbf{h}_n$, an estimated hidden vector $\widetilde{\mathbf{h}}_n$ is computed

$$\widetilde{\mathbf{h}}_n = \tanh(\mathbf{W}_c[\mathbf{c}_n, \mathbf{h}_n]), \tag{67}$$

and the upcoming calculation at this iteration relies on this final estimated hidden vector $\widetilde{\mathbf{h}}_n$. This procedure is illustrated in Fig.7.

D. Extension of Self-Correction Modular by Steepest Descent

The convergence of SAD partially depends on its self-correction modular, which is realized by GD method. Henceforth, by extending it to other iterative methods that converge faster, an improved convergence of SAD can be achieved. Here we select the steepest descent (STD) method, which is a special case of GD, as an instance here to verify the benefit of predefined step-sizes in SAD.

Considering the objective function $h(\mathbf{x})$ in (27), by setting its derivative with respect to the step size δ to zero, δ_j that minimizes the updated value $h(\mathbf{x}_j - \delta\nabla h(\mathbf{x}_j))$ at j-th step is given by

$$\delta_j = \frac{\nabla h(\mathbf{x}_j)^T \nabla h(\mathbf{x}_j)}{\nabla h(\mathbf{x}_j)^T \mathbf{A} \nabla h(\mathbf{x}_j)}. \tag{68}$$

Substituting this step-size δ_j into the updating equation in (28) gives the STD method. Following the same definition of the weighted norm in (29), it has been derived in [38] that STD converges obeying

$$\|\mathbf{x}_{j+1} - \mathbf{x}^*\|_{\mathbf{A}}^2 \leq \left(\frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1}\right)^2 \|\mathbf{x}_j - \mathbf{x}^*\|_{\mathbf{A}}^2. \tag{69}$$

Hence, similar to our previous derivation, a lower bound of the number of STD steps is given by

$$m \geq \frac{\frac{1}{2}\ln\frac{\epsilon}{r}}{\ln\frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1}} = \log_{\beta_{\mathrm{STD}}}\frac{r}{\epsilon}, \tag{70}$$

where $\beta_{\mathrm{STD}} = (\frac{\lambda_n + \lambda_1}{\lambda_n - \lambda_1})^2 > 1$ is the base with respect to STD method. Again, owing to (52), it is easy to arrived at the result that the STD base β_{STD} here is exactly the antenna ratio, $\beta_{\mathrm{STD}} = \alpha$. For most cases, the STD base is much greater than the GD base, $\beta_{\mathrm{STD}} > \beta_{\mathrm{GD}}$. Therefore, in GPGD method, the lower bound of m_{STD} for a SAD using STD steps in the self-correction modular is smaller than that of SAD using GD steps, m_{GD}, leading to a faster convergence. On the other hand, in DetNet, instead of using the trainable step-size $\boldsymbol{\delta}$, the determinant step-size can be calculated by this STD iteration. It would be shown in our simulation that the latter STD version of DetNet shows a better performance than its original one. This verifies that setting the step-size δ as trainable parameters is not an optimal choice. Therefore our attempt to predefine δ rather than training it makes sense.

VI. Numerical Results

During the training phase, in accordance with the analysis under back-propagation through time (BPTT) in [39], we adopt the weighted loss function

$$\min \sum_{n=1}^{n=P} \log(n) f_{\mathbf{x}_t}(\widetilde{\mathbf{x}}_n) \tag{71}$$

to involve each iteration in BP process as much as possible and make the vanishing gradient problem less likely to occur. The training procedure works on the DL library PyTorch. We draw on 150,000 to train SAD with Adam Optimizer [40] using a batch size of 100. The noise variance is randomly generated within Eb/N0 $\sim \mathcal{U}([\text{Eb/N0}]_{\min} - 1, [\text{Eb/N0}]_{\max} + 1)$, where

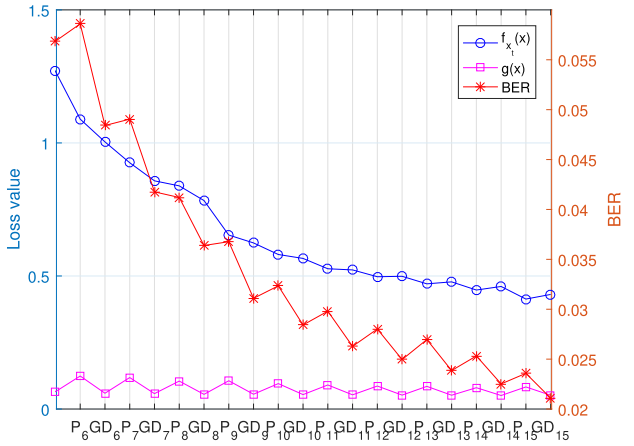

Fig. 8. Monitored $f_{\mathbf{x}_t}(\mathbf{x})$ and $g(\mathbf{x})$ value in DetNet, as well as the BER after each projection and GD step.

$[\cdot]_{\min}$ and $[\cdot]_{\max}$ take the minimal and maximal values used in detection. The learning rate is set as 0.001 and would decay by 0.97 after each epoch. We point out that in the previous analysis we focus on the comparison to the ZF detector for simplicity consideration, but in the following experiments, the augmented $\underline{\mathbf{H}}$ and $\underline{\mathbf{y}}$ are utilized in all detectors for a fair comparison. The source code is available at *https://github.com/LanxinHe/SAD*.

The bit error rate (BER) performance is presented with all the experiments conducted assuming i.i.d Rayleigh MIMO channels. In Fig.8 and Fig.9, we present the monitored loss value in DetNet and SAD respectively, where the adversarial behavior of $f_{\mathbf{x}_t}(\mathbf{x})$ and $g(\mathbf{x})$ at the stage II of PGD can be observed. Notice that in DetNet (Fig.8), the BER would increase after each projection, which means that the detection deteriorates after the projection in original PGD-detection. This is in accordance with our analysis about the mistakes introduced by projection, and it is clear that one single GD iteration is not enough for correcting these mistakes, leading to the computation waste by projection. However, for SAD, this waste of computing resource has been solved thanks to the adoption of GPGD method. Also recall the principle of selecting m we proposed in Sec.IV-A. Both of these two conditions can be interpreted as in Fig.9, where the dashed blue line shows an expected trend of $f_{\mathbf{x}_t}$ indicated by (43a), and (43b) implies a narrow distance between the purple squares at each GD stage and the dashed purple line. By cooperating the projection with multiple times GD, the potential of both these two operations can be exploited.

Fig.10 verifies our step-size condition in (44). With respect to the 16×16, QPSK scheme in the left of picture, according to the condition (44), the suitable step-size lies between 0 and 0.0156. As the result shows, utilizing the two step-sizes (0.012 and 0.014) that obey this condition, the GD method performs in an ideal way with the tendency converging to MMSE detection. However, if a step-size that locates even a little bit outside the range (0.016) is adopted, GD method begins to deteriorate. Meanwhile, an extreme case with $\alpha = 30$ is shown in the right picture, where the two terms in the condition (44) are 0.006 and 0.0052 respectively. It can be seen that using $\delta = 0.005$, the GD method satisfyingly converges to MMSE detection, while once this step-size increases to

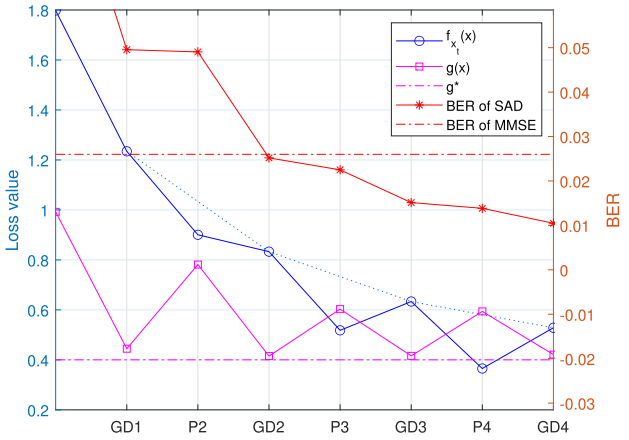

Fig. 9. Monitored $f_{\mathbf{x}_t}(\mathbf{x})$ and $g(\mathbf{x})$ value in SAD, as well as the BER after each projection and GD step.

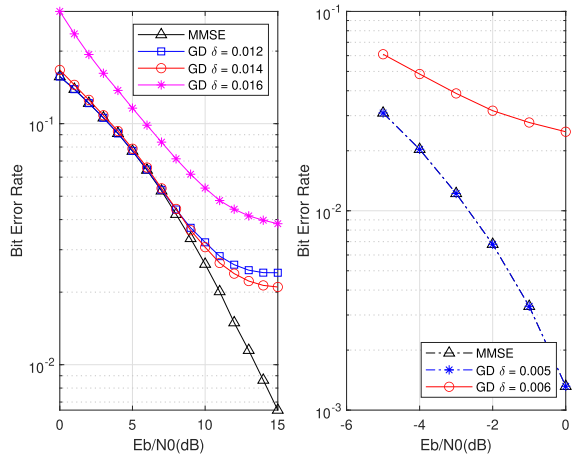

Fig. 10. Verification of the condition (44), where the left picture shows the performance under 16×16, QPSK, and the right shows the performance under 120×4, 64-QAM. All the GD methods iterate 50 times.

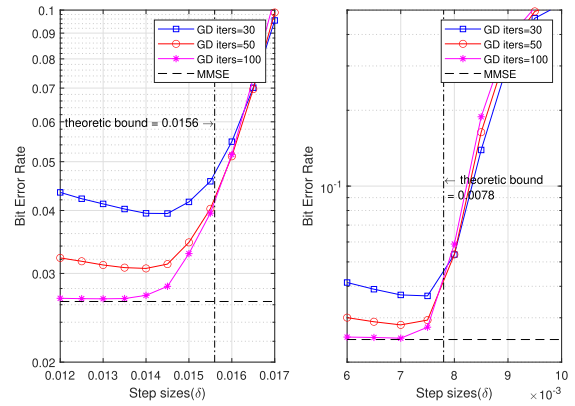

Fig. 11. The performance of GD iterations by varying step-sizes δ, with the SNR fixed at Eb/N0 = 10dB. The left picture corresponds to the case $N = K = 16$, where the theoretic bound computed according to the condition (44) is 0.0156, and the right refers to the case $N = K = 32$ with the related theoretic bound $\delta < 0.0078$.

$\delta = 0.006$, it deteriorates again. Even this situation rarely happens in practice, it demonstrates our proposed step-size condition.

Furthermore, we also explore the performance of GD iterations with varying step-sizes δ and fixed SNR in Fig.11. We can see that the theoretic bound derived in (44) does give a near-optimal upper bound on the selection of step-size. When the GD algorithm iterates within the small number

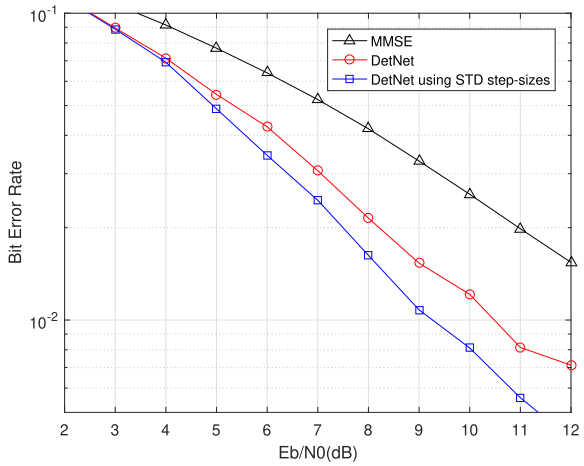

Fig. 12. Replacing the trainable step-sizes in DetNet by those calculated using STD method: under 16×16, QPSK.

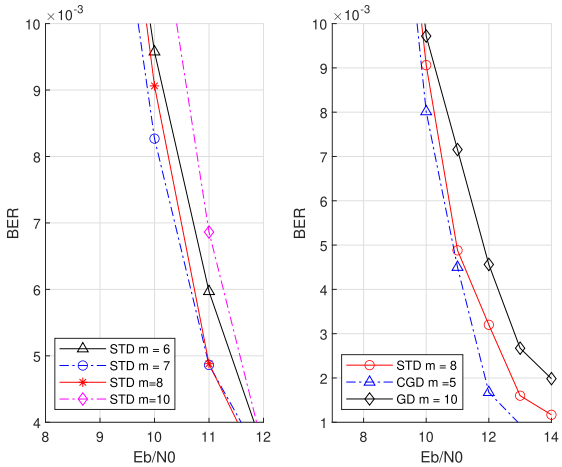

Fig. 13. Performance comparison among STD-SADs with different m (left); SADs taking different line search methods in self-correction modular (right).

of iterations (like 30 and 50 in Fig.11), which is often the case taking place in the PGD method, either choosing a δ too small or too large fails to achieve the potential of GD method. Hence it is necessary to bound the step-size inside the proposed condition. Otherwise, the iteration of GD would become quite ineffective. Besides, as the number of antennas goes larger, the upper bound given in (44) becomes a more accurate one. This is owing to the fact that the approximation in (52) performs more precisely when N goes larger.

A big difference between SAD and other detection networks is that we do not set the step-size δ as trainable. This may cause some confusions, so we here make use of STD method to verify our proposition. For DetNet, instead of training its step-sizes, we manage to calculate the related STD step-sizes and plot the performance of these two trained models in Fig.12. From it, we can see that without training the step-sizes, DetNet even shows a better performance. Note that when choosing to train step-sizes, what is paid is more than just the computation resource used to learn step-sizes. It is the sacrificed attention that could have been put on other learnable parameters. Hence, in SAD, we argue to predefine the step-sizes rather than training them, and its optimal choice discussed in Sec.IV-B shows a great importance.

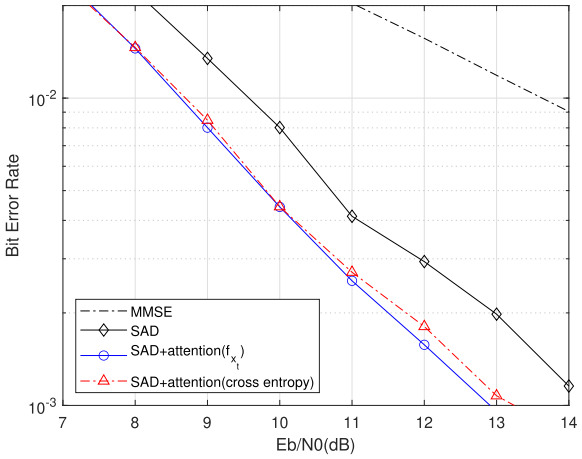

Fig. 14. Performance gain after adopting attention mechanism in SAD.

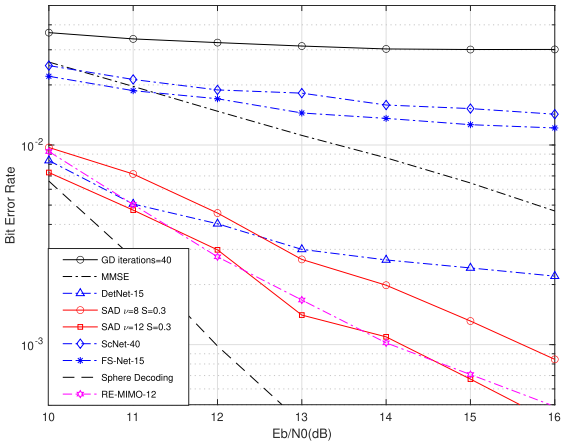

Fig. 16. Comparison of BER performance for 16×16 MIMO system using QPSK, all related step sizes δ are set as 0.012.

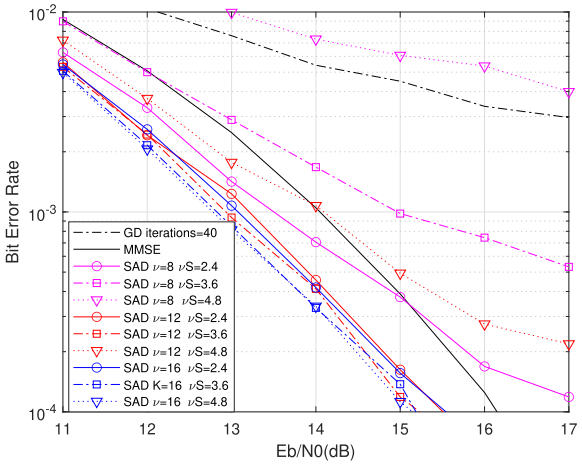

Fig. 15. Comparison of BER performance for 30×20 MIMO system using 16-QAM, all related step sizes δ are set as 0.008.

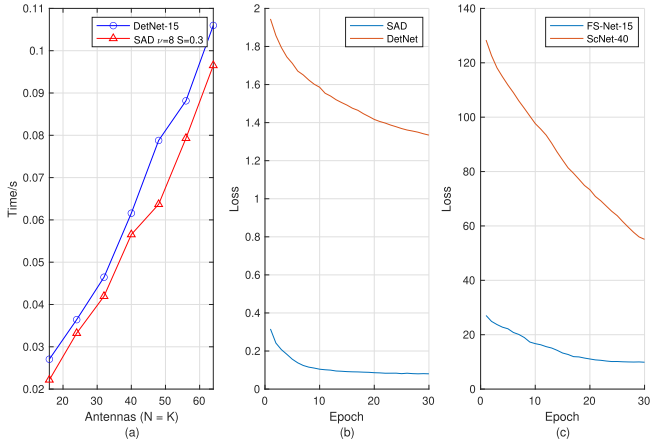

Fig. 17. (a): Running time comparison between DetNet and SAD. (b) and (c): Training loss of SAD, DetNet, FS-Net and ScNet.

In Fig.13 we train the SAD and its STD-extended version with different m (in the left), and compare the performance of SADs taking different line search methods in self-correction modular (in the right). For each SAD, either a smaller or a greater m may influence the final performance, implying that there is a suitable m lying between the principle of selecting m in Sec.IV-A. In the right the CGD method is included as well. As a nonlinear method, CGD has the fastest convergence rate among these three presented methods. Accordingly, the corresponding suitable self-correction iterations m of these three detectors decrease as their self-correction modulars converge faster, $m_{\mathrm{GD}} > m_{\mathrm{STD}} > m_{\mathrm{CGD}}$. Even β_{CGD} is difficult to develop, we can see from $\beta_{\mathrm{STD}} > \beta_{\mathrm{GD}}$ that the STD-version SAD owns a smaller lower bound on m, thus just requiring a smaller m to reach its best performance. Besides, it is interesting to see that using a faster-converging iterative method in self-correction modular helps the SAD to resist the performance deterioration in higher Eb/N0 range.

Fig.14 demonstrates the performance gain of the adoption of attention mechanism. We train the attention-mechanism-augmented SAD under two different loss functions, $f_{\mathbf{x}_t}(\mathbf{x})$ and the converted cross-entropy loss f_{equi}, respectively. The SAD trained with both of these two loss functions outperform the standard SAD, demonstrating that maintaining the mutual information between estimates and hidden vectors does boost the decrease of loss, so that the detection become more efficient. Here we adopt the scoring function in (64) and there are other alternative forms such as inner product and multilayer perception (MLP). Meanwhile, the two SADs trained under these two different loss functions show little difference, verifying the equivalence between $f_{\mathbf{x}_t}(\mathbf{x})$ and f_{equi} as we discussed in Sec.V-C.

In Fig.15 we present the performance of SADs with different coefficient ν and sparsity S under 30×20 16-QAM system, where νS can be viewed as a metric of the space complexity of each SAD. For example, since $S = 0.3$ means only 30% of the weights are nonzero and all others are zero-valued, $\nu S = 8 \times 0.3 = 2.4$ conveys that the original dense size of SAD has been reduced from $8K$ to $2.4K$. It can be concluded that BER drops rapidly as the underlying dimensionality νK increases or as the model turns more sparse, but this improvement would saturate. Therefore, even the training burden is affordable, we do not need to set the dimension of hidden vector too high just in order to improve performance. For instance, an ideal set under this case would be $\nu = 12$ rather than $\nu = 16$ for complexity consideration.

Fig.16 compares the SAD with other PGD-based detectors where a 15-layers DetNet, a 15-layers FS-Net and a 40-layers ScNet are involved. The latter two focus on reducing their complexity by applying only element-wise product. It can be

seen that among these four detectors, the pure SAD achieves the best performance, and it is worth pointing out that the DetNet, which even requires the largest amount of trainable parameters, performs less competitively than SAD. What is more, during detection, the SAD performs the projection with least times (4-times projection compared to 15-times in DetNet, 15-times in FS-Net and 40-times in ScNet). Therefore, we can conclude that by unfolding GPGD method, the SAD works more efficient and achieves a better trade-off between complexity and detection performance. Besides, the performance of another DL-based MIMO detector, RE-MIMO, which is established based on the RIM and adopts the transformer, is presented as well. It can be seen that the SAD performing projection 4 times shows the comparable performance with a 12-layers RE-MIMO detector.

In Fig.17, the running time and real-time training loss versus epochs are presented. The DetNet and SAD in Fig.17(a) perform detection with the same batch size of 20. From this it can be seen that even GPGD method would execute more GD steps in total than the original PGD method does, the former can achieve better performance with less complexity. The training loss comparisons between SAD and other three DNN-based detector are presented in Fig.17(b) and Fig.17(c), where all the detectors are trained with the same number of training set, 150000, per epoch. Note that due to the loss function that are utilized by each detector are distinct, comparing their absolute value is not a wise option. However, the convergence during training can be told by the tendency of their related loss versus epochs. Both SAD and FS-Net reach their loss floor within 30 epochs while DetNet and ScNet still show a decreasing tendency, and the proposed SAD converges even faster than the FS-Net. This reflects that the SAD is easy to train and shows a faster convergence during training.

VII. Conclusion

In this paper, the PGD-based detector has been studied to achieve the performance improvement in massive MIMO systems. By adjusting the projection and GD steps in a more flexible way, their corresponding loss functions manage to strike a balance, leading to the proposed GPGD method, so that the potential of PGD method can be exploited. Specifically, after analyzing the behaviour of projection and GD step, we develop the suitable choice for the steps m and the step-size δ, thus resulting in a more promising and stable detection. Furthermore, as one realization of GPGD method, the SAD, has been proposed under the help of DNN to achieve a better decoding performance. Meanwhile, we integrate the attention mechanism into SAD and extend it by STD for performance improvement and efficiency. Finally, simulation results based on the massive MIMO detection demonstrate the advantage of GPGD method and the system gain in terms of performance.

References

[1] F. Tariq, M. R. A. Khandaker, K. Wong, M. A. Imran, M. Bennis, and M. Debbah, "A speculative study on 6G," *IEEE Wireless Commun.*, vol. 27, no. 4, pp. 118–125, Aug. 2020.

[2] W. Jiang, B. Han, M. A. Habibi, and H. D. Schotten, "The road towards 6G: A comprehensive survey," *IEEE Open J. Commun. Soc.*, vol. 2, pp. 334–366, 2021.

[3] E. G. Larsson, O. Edfors, F. Tufvesson, and T. L. Marzetta, "Massive MIMO for next generation wireless systems," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 186–195, Feb. 2014.

[4] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge, U.K.: Cambridge Univ. Press, 2005.

[5] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger, "Closest point search in lattices," *IEEE Trans. Inf. Theory*, vol. 48, no. 8, pp. 2201–2214, Aug. 2002.

[6] M. O. Damen, H. El Gamal, and G. Caire, "On maximum-likelihood detection and the search for the closest lattice point," *IEEE Trans. Inf. Theory*, vol. 49, no. 10, pp. 2389–2402, Oct. 2003.

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

[8] F. Emmert-Streib, Z. Yang, H. Feng, S. Tripathi, and M. Dehmer, "An introductory review of deep learning for prediction models with big data," *Frontiers Artif. Intell.*, vol. 3, pp. 1–12, Feb. 2020.

[9] N. Buduma and N. Locascio, *Fundamentals of Deep Learning: Designing Next-Generation Machine Intelligence Algorithms*. Sebastopol, CA, USA: O'Reilly Media, 2017.

[10] J. Wang, C. Jiang, H. Zhang, Y. Ren, K. Chen, and L. Hanzo, "Thirty years of machine learning: The road to Pareto-optimal wireless networks," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 1472–1514, 3rd Quart., 2020.

[11] D. W. Otter, J. R. Medina, and J. K. Kalita, "A survey of the usages of deep learning for natural language processing," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 2, pp. 604–624, Feb. 2021.

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

[13] N. Samuel, T. Diskin, and A. Wiesel, "Learning to detect," *IEEE Trans. Signal Process.*, vol. 67, no. 10, pp. 2554–2564, May 2019.

[14] G. Gao, C. Dong, and K. Niu, "Sparsely connected neural network for massive MIMO detection," in *Proc. IEEE 4th Int. Conf. Comput. Commun. (ICCC)*, Dec. 2018, pp. 397–402.

[15] N. T. Nguyen and K. Lee, "Deep learning-aided Tabu search detection for large MIMO systems," *IEEE Trans. Wireless Commun.*, vol. 19, no. 6, pp. 4262–4275, Jun. 2020.

[16] J. Yang and H. Li, "Deep learning based detection and channel tracking for MIMO systems," in *Proc. 5th Int. Conf. Comput. Commun. Syst. (ICCCS)*, May 2020, pp. 635–639.

[17] A. Mohammad, C. Masouros, and Y. Andreopoulos, "Complexity-scalable neural-network-based MIMO detection with learnable weight scaling," *IEEE Trans. Commun.*, vol. 68, no. 10, pp. 6101–6113, Oct. 2020.

[18] J. Sun, Y. Zhang, J. Xue, and Z. Xu, "Learning to search for MIMO detection," *IEEE Trans. Wireless Commun.*, vol. 19, no. 11, pp. 7571–7584, Nov. 2020.

[19] M. Mohammadkarimi, M. Mehrabi, M. Ardakani, and Y. Jing, "Deep learning-based sphere decoding," *IEEE Trans. Wireless Commun.*, vol. 18, no. 9, pp. 4368–4378, Sep. 2019.

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

[21] Z. Wang, R. M. Gower, Y. Xia, L. He, and Y. Huang, "Randomized iterative methods for low-complexity large-scale MIMO detection," *IEEE Trans. Signal Process.*, vol. 70, pp. 2934–2949, 2022.

[22] Y. Wei, M.-M. Zhao, M. Hong, M.-J. Zhao, and M. Lei, "Learned conjugate gradient descent network for massive MIMO detection," *IEEE Trans. Signal Process.*, vol. 68, pp. 6336–6349, 2020.

[23] H. He, C. Wen, S. Jin, and G. Y. Li, "Model-driven deep learning for MIMO detection," *IEEE Trans. Signal Process.*, vol. 68, pp. 1702–1715, 2020.

[24] X. Tan et al., "Improving massive MIMO message passing detectors with deep neural network," *IEEE Trans. Veh. Technol.*, vol. 69, no. 2, pp. 1267–1280, Feb. 2020.

[25] H. Chen, G. Yao, and J. Hu, "Algorithm parameters selection method with deep learning for EP MIMO detector," *IEEE Trans. Veh. Technol.*, vol. 70, no. 10, pp. 10146–10156, Oct. 2021.

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

[27] M. Kim and D. Park, "Learnable MIMO detection networks based on inexact ADMM," *IEEE Trans. Wireless Commun.*, vol. 20, no. 1, pp. 565–576, Jan. 2021.

[28] P. L. Combettes and J.-C. Pesquet, "Proximal splitting methods in signal processing," in *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*. Cham, Switzerland: Springer, 2011, pp. 185–212.
[29] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
[30] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 2012.
[31] S. Wright et al., "Numerical optimization," *Springer Sci.*, vol. 35, nos. 67–68, p. 7, 1999.
[32] A. Edelman, "Eigenvalues and condition numbers of random matrices," *SIAM J. Matrix Anal. Appl.*, vol. 9, no. 4, pp. 543–560, Oct. 1988.
[33] S. Ahmad and L. Scheinkman, "How can we be so dense? The benefits of using highly sparse representations," 2019, *arXiv:1903.11257*.
[34] L. He, T. Liu, and Z. Wang, "Recurrent sparse MIMO detection network based on modified projected gradient descent method," in *Proc. IEEE 94th Veh. Technol. Conf. (VTC-Fall)*, Sep. 2021, pp. 1–5.
[35] D. Dai, L. Yu, and H. Wei, "Parameters sharing in residual neural networks," *Neural Process. Lett.*, vol. 51, no. 2, pp. 1393–1410, Apr. 2020.
[36] S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," *Int. J. Uncertainty, Fuzziness Knowl.-Based Syst.*, vol. 6, no. 2, pp. 107–116, Apr. 1998.
[37] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proc. 25th Int. Conf. Mach. Learn.*, 2008, pp. 1096–1103.
[38] D. G. Luenberger, *Introduction to Linear and Nonlinear Programming*. New York, NY, USA: Springer, 2008.
[39] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 1310–1318.
[40] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

Lanxin He (Graduate Student Member, IEEE) received the B.S. degree from the College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics (NUAA), Nanjing, China, in 2021, where she is currently pursuing the M.S. degree. Her research interests include machine learning and deep learning applications on uplink signal detection of massive MIMO systems.

Zheng Wang (Senior Member, IEEE) received the B.S. degree in electronic and information engineering from the Nanjing University of Aeronautics and Astronautics (NUAA), Nanjing, China, in 2009, the M.S. degree in communications from The University of Manchester, Manchester, U.K., in 2010, and the Ph.D. degree in communication engineering from Imperial College London, U.K., in 2015.

From 2015 to 2016, he was a Research Associate with Imperial College London. From 2016 to 2017, he was a Senior Engineer with the Radio Access Network Research and Development Division, Huawei Technologies Company. From 2017 to 2020, he was an Associate Professor with the College of Electronic and Information Engineering, NUAA. Since 2021, he has been an Associate Professor with the School of Information and Engineering, Southeast University, Nanjing. His current research interests include massive MIMO systems, machine learning and data analytics over wireless networks, and lattice theory for wireless communications.

Shaoshi Yang (Senior Member, IEEE) received the B.Eng. degree in information engineering from the Beijing University of Posts and Telecommunications (BUPT), China, in 2006, and the Ph.D. degree in electronics and electrical engineering from the University of Southampton, U.K., in 2013. From 2008 to 2009, he was a Researcher of WiMAX standardization with Intel Labs China. From 2013 to 2016, he was a Research Fellow with the School of Electronics and Computer Science, University of Southampton. From 2016 to 2018, he was a Principal Engineer with Huawei Technologies Company Ltd., where he made substantial contributions to the company's products and solutions on 5G base stations, wideband IoT, and cloud gaming/VR. He is currently a Full Professor with BUPT. His research interests include 5G wireless networks, massive MIMO, iterative detection and decoding, mobile ad hoc networks, distributed artificial intelligence, and cloud gaming/VR. He is a Guest Researcher with the Isaac Newton Institute for Mathematical Sciences, Cambridge University. He was a recipient of the Dean's Award for Early Career Research Excellence from the University of Southampton in 2015, the Huawei President Award of Wireless Innovations in 2018, the IEEE Technical Committee on Green Communications and Computing (TCGCC) Best Journal Paper Award in 2019, and the IEEE Communications Society Best Survey Paper Award in 2020. He is an Editor of IEEE SYSTEMS JOURNAL, IEEE WIRELESS COMMUNICATIONS LETTERS, and *Signal Processing* (Elsevier). He was an invited international reviewer of the Austrian Science Fund (FWF). For more details of his research progress, please refer to https://shaoshiyang.weebly.com/

Tao Liu received the B.Sc. and Ph.D. degrees in electrical engineering from Xi'an Jiaotong University, Xi'an, China, in 2004 and 2013, respectively. He is currently with the College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, China. His research interests include cooperative communication systems, synchronization, and channel estimation for wireless communication systems.

Yongming Huang (Senior Member, IEEE) received the B.S. and M.S. degrees from Nanjing University, Nanjing, China, in 2000 and 2003, respectively, and the Ph.D. degree in electrical engineering from Southeast University, Nanjing, in 2007.

Since March 2007, he has been a Faculty Member with the School of Information Science and Engineering, Southeast University, where he is currently a Full Professor. From 2008 to 2009, he was with the Signal Processing Laboratory, Royal Institute of Technology, Stockholm, Sweden. He has authored or coauthored more than 200 peer-reviewed papers, and holds more than 80 invention patents. His research interests include intelligent 5G/6G mobile communications and millimeter wave wireless communications. He submitted around 20 technical contributions to IEEE standards. He was awarded a certificate of appreciation for outstanding contribution to the development of IEEE standard 802.11aj. He was an Associate Editor of the IEEE TRANSACTIONS ON SIGNAL PROCESSING and a Guest Editor of the IEEE JOURNAL SELECTED AREAS IN COMMUNICATIONS. He is currently an Editor-at-Large of the IEEE OPEN JOURNAL OF THE COMMUNICATIONS SOCIETY and an Associate Editor of the IEEE WIRELESS COMMUNICATIONS LETTERS.