Fast Online Channel Estimation in Massive MIMO: A Zero-Shot Self-Supervised Approach

Zijun Gao, *Graduate Student Member, IEEE,* Wenqiang Yi, *Member, IEEE,* Fatma Benkhelifa, *Senior Member, IEEE,* and Arumugam Nallanathan, *Fellow, IEEE*

***Abstract*—With the growing number of antennas in massive multiple-input multiple-output (MIMO) systems, robust and fast channel estimation becomes increasingly critical yet remains highly challenging. In this work, we propose a lightweight zero-shot self-supervised (ZS-SS) learning framework. It leverages non-local self-similarity in wireless channels to construct a channel-coefficient bank and generate training pairs via randomized and non-contiguous spatial permutations to decorrelate noise. These pairs then train a compact convolutional neural network (CNN) with a specially designed composite loss for robust channel estimation. To further improve adaptability and efficiency, we incorporate a meta-learning approach for fast inference time to dynamic channel environments. Simulations under Gaussian and representative non-Gaussian scenarios show that our method achieves up to 90% gains over traditional estimators and consistent improvements over state-of-the-art baselines, while running nearly 100 times faster. This demonstrates its practicality and suitability for real-time deployment in resource-limited massive MIMO systems.**

***Index Terms*—Channel Estimation, Massive MIMO, Meta-SGD, Non-Gaussian Noise, Zero-Shot Self-Supervised Learning.**

I. Introduction

Massive multiple-input multiple-output (MIMO) technology has emerged as a cornerstone of modern wireless communication systems, providing substantial enhancements in spectral efficiency, network throughput, and reliability through spatial multiplexing and spatial diversity [2]–[5]. The effectiveness of massive MIMO systems critically depends on the acquisition of accurate channel state information (CSI), which forms the basis for essential downstream tasks including precoding, beamforming, resource allocation, and signal detection [6]–[8]. Despite its importance, channel estimation in massive MIMO remains challenging due to high-dimensional channel matrices, significant pilot overhead, severe computational complexity, and susceptibility to various noise and interference sources encountered in practical communication environments [9]–[11]. Traditional methods, such as least squares (LS) and minimum mean squared error (MMSE), are widely adopted due to their simplicity and solid theoretical foundations but suffer from critical limitations [12]. Nonetheless, these traditional estimators fundamentally rely on prior knowledge or assumptions about channel statistics or environmental characteristics. For instance, LS estimators, while computationally efficient, often yield inaccurate estimates under low signal-to-noise ratio (SNR) and severe interference conditions, negatively impacting estimation performance [13]. MMSE estimators offer theoretically optimal linear solutions but demand precise knowledge of channel correlation statistics, typically requiring extensive sampling and frequent updates [14]. Such stringent requirements severely constrain their practicality, especially in dynamically changing propagation environments under real-world massive MIMO deployments.

Unlike traditional estimation techniques, deep learning (DL)-based methods inherently leverage patterns extracted from training samples rather than explicit prior statistical knowledge of the channel [15]–[17]. Rather than relying on explicit statistical modeling of the channel, DL-based approaches automatically learn the complex mapping from received signals to channel coefficients directly from data, without requiring prior knowledge of the underlying distribution [18]. Consequently, DL methods exhibit better adaptability to non-Gaussian, time-varying, or otherwise unknown channel conditions, where traditional estimators often fail due to the theoretical model mismatch [19]–[21]. This adaptability makes DL especially well-suited for the dynamic and complex environments, such as massive MIMO communications. These strengths motivate further exploration of advanced DL-based channel estimation methods tailored to practical deployment.

A. Related Work and Motivation

Recently, DL-based methods have attracted significant attention in tackling channel estimation tasks [20]–[23]. For instance, the ChannelNet [24] utilizes a denoising convolutional neural network (DnCNN) structure. It models the channel's time-frequency responses as a two-dimensional image, and reconstructs the complete CSI from sparse pilot measurements via image super-resolution and restoration techniques. Similarly, the NDR-Net [25] also incorporated a DnCNN structure for channel estimation tasks under unknown noise conditions. Despite promising accuracy, these approaches predominantly rely on supervised learning frameworks. They require large labeled datasets, extensive training times, and substantial computational resources. Collecting a large set of

Part of this work has been accepted by the 2025 IEEE Global Communications Conference (GLOBECOM) [1].

Z. Gao and F. Benkhelifa are with the School of Electronic Engineering and Computer Science, Queen Mary University of London, E1 4NS, London, U.K. (e-mails: {zijun.gao, f.benkhelifa}@qmul.ac.uk).

W. Yi is with the School of Computer Science and Electronic Engineering, University of Essex, CO4 3SQ, Colchester, U.K. (e-mail: w.yi@essex.ac.uk).

A. Nallanathan is with the School of Electronic Engineering and Computer Science, Queen Mary University of London, E1 4NS, London, U.K. and also with the Department of Electronic Engineering, Kyung Hee University, Yongin-si, Gyeonggi-do 17104, Korea (e-mail: a.nallanathan@qmul.ac.uk).

labeled data is costly in realistic wireless systems. Moreover, their practicality remains limited under dynamically varying channel conditions. Frequent offline retraining under new environments further increases overhead and latency, complicating practical deployment.

To mitigate these limitations, recent efforts have explored self-supervised methods for channel estimation. Balevi et al. [14] introduced a deep image prior (DIP)-inspired neural network estimator. It first denoises the received signals and subsequently applies a conventional LS estimator, optimizing network parameters dynamically per channel realization without labeled data. Kim et al. [12] integrated a DIP denoising step into channel prediction. This approach cleans the training data and notably boosting accuracy at low SNR. Their predictor benefits primarily from this DIP-based pre-processing rather than heavy supervised retraining. More recently, Kang et al. [26] proposed a Self2Self solution, which generates self-supervised training labels via Bernoulli sampling. It further employs blind-spot strategies and dropout techniques, reportedly outperforming DIP-based approaches. Despite recent advancements, critical challenges remain unresolved. Existing approaches typically require large-scale neural networks, incurring substantial computational costs and prolonged run times, limiting real-time deployment in resource-constrained, dynamic wireless scenarios. Thus, developing lightweight models with reduced computational complexity and enhanced generalization and adaptability to varying channel conditions remains essential. This underscores the need for efficient and practical solutions tailored for realistic massive MIMO systems.

In addition to the algorithm design, the real-time estimation solution needs to adapt to the practical environment, especially for practical noise. Most existing solutions are designed based on additive white Gaussian noise (AWGN), which limits their adaptivity [27]–[30]. In reality, wireless communication signals are contaminated by not only Gaussian but also non-Gaussian noise, such as impulsive noise (IN). This complicates the design of optimal estimation algorithms due to uncertainties in the noise prior and inevitable parameter estimation errors [31], [32]. For non-Gaussian with IN, traditional solutions based on AWGN have degraded performance as they commonly have weaker denoising ability for uneven noise [33]. A few recent works have made efforts in this regard. For instance, Zhao et al. [34] proposed an end-to-end transceiver for Middleton/α-stable channels, using an impulsive generative adversarial network (GAN) and wavelet-based decoder to improve bit error rate (BER) under strong impulses. Sadeghi et al. [35] tackled similar issues in power line communications (PLC) with a three-stage deep neural network (DNN) combining denoising and cascaded estimation. However, both approaches rely on heavyweight, task-specific neural architectures. More specifically, the authors in [34] merge estimation, equalization, and detection into a single pipeline without providing stand-alone CSI. The model in [35] was tailored for PLC noise, which is different from that in massive MIMO systems.

Therefore, there is a pressing need for developing a lightweight, adaptive channel estimation framework that achieves consistently high performance across both Gaussian and diverse non-Gaussian noise scenarios. Such a framework should not rely on large labeled datasets and explicit prior knowledge of the noise distribution, enabling practical, real-time deployment in dynamically changing massive MIMO environments.

B. Contributions

Motivated by the aforementioned challenges, this paper proposes a lightweight zero-shot self-supervised (ZS-SS) learning framework that integrates a meta-learning approach with stochastic gradient descent (Meta-SGD) for massive MIMO channel estimation. The main contributions are:

- We propose a ZS-SS learning framework for massive MIMO channel estimation without any labeled data, applicable to both Gaussian and non-Gaussian noise. Motivated by the challenge of generating self-supervised pairs from only a single noisy observation, we exploit intrinsic non-local self-similarity and spatially repeated patterns in large antenna arrays. Specifically, we construct a channel-coefficient bank from LS pre-estimate, and generate pseudo-training pairs via randomized, non-contiguous spatial permutations, effectively decorrelating noise.
- We employ a compact four-layer CNN as the estimation model, significantly reducing computational complexity compared to large-scale architectures like U-Net [26], [36], requiring only about one-tenth the trained parameters. To further accelerate convergence and enhance generalization, we design a novel composite loss function combining cross-reconstruction and sampling-consistency terms. Unlike existing self-supervised approaches, our method effectively prevents overfitting and improves robustness across various noise distributions.
- We further integrate Meta-SGD, a meta-learning approach enabling element-wise adaptive learning rates, significantly reducing online adaptation time and alleviating overfitting. Compared with conventional optimization strategies, Meta-SGD inherently supports online adaptation, enabling continuous tracking of evolving channel conditions during real-time deployments.
- Simulations under Gaussian and representative non-Gaussian scenarios demonstrate superior robustness, with performance gains up to 90% over the LS estimators and consistent improvement of $15\% - 35\%$ over state-of-the-art deep learning methods. In addition, our framework is extremely lightweight ($225\,\mathrm{k}$), requires minimal runtime memory ($130\,\mathrm{MB}$), and achieves nearly 100 times faster run time than existing methods on both CPU and GPU settings.

C. Organization

The paper is organized as follows. Section II presents the system model and problem formulation. Section III describes the proposed ZS-SS channel estimation method. Section IV details the Meta-SGD algorithm. Section V provides simulation results and performance comparisons. Section VI

provides further discussions and directions for future work. Finally, Section VII concludes the paper.

II. System Model and Problem Formulation

In this section, we first introduce the employed noise models, then formulate the pilot-aided channel estimation problem in massive MIMO system. Lastly, we provide the framework for our ZS-SS learning approach as shown in Fig. 1.

A. Noise Model

In practical wireless communication systems, the additive interference at the receiver often consists of multiple simultaneous noise processes, typically comprising both Gaussian and non-Gaussian components [27], [30]. Specifically, thermal background noise is conventionally modeled as AWGN, characterized by its simplicity and well-understood statistical properties. In contrast, non-Gaussian disturbances, arising from factors such as electromagnetic interference, impulse phenomena, or heavy-tailed processes [37]–[39], pose additional challenges due to their unpredictable and complex statistical characteristics [31], [32]. To reflect this realistic scenario, we conceptually model the received noise as a general additive combination:

$$\mathbf{n} = a\,\mathbf{n}_{\mathrm{G}} + b\,\mathbf{n}_{\mathrm{NG}}, \tag{1}$$

where $\mathbf{n}_{\mathrm{G}} \sim \mathcal{CN}(\mathbf{0}, \sigma^2\mathbf{I})$ represents the Gaussian noise component, $\mathbf{n}_{\mathrm{NG}}$ denotes an arbitrary non-Gaussian noise term, and coefficients $a, b \geq 0$ reflect the relative strength of these noise sources. Here, σ^2 denotes the noise power and $\mathbf{I}$ is the identity matrix of appropriate dimension. The distribution of $\mathbf{n}$ is denoted by $f_n(a, b)$.

It is worth emphasizing that our proposed estimation method is explicitly designed without reliance on specific assumptions or prior knowledge about the noise parameters or distributions. Thus, it inherently maintains generality and robustness across diverse interference environments. Therefore, it could effectively address scenarios involving both purely Gaussian conditions ($b = 0$), purely non-Gaussian conditions ($a = 0$), or their combinations. In our experimental evaluations, we specifically consider representative non-Gaussian noise models, including Laplacian and Bernoulli-Gaussian impulsive noise (BGIN), to rigorously assess the robustness and generalization performance of our approach. Detailed descriptions of these noise cases and their experimental setups are provided in Section V.

B. Signal Model

We consider a narrowband single-user massive MIMO system operating in time-division duplexing (TDD). The base station (BS) and the user equipment (UE) employ n_{rx} and n_{tx} antennas, respectively. The UE sends orthogonal pilot signals represented by

$$\mathbf{U} \in \mathbb{C}^{n_{\mathrm{tx}} \times u}, \quad \mathbf{U}\mathbf{U}^H = u\,\mathbf{I}_{n_{\mathrm{tx}}}, \tag{2}$$

where u is the length of the pilot sequence. We consider scenarios where pilots are perfectly orthogonal, including typical single-cell multi-user systems or multi-cell multi-user environments where pilots are designed to be orthogonal. Assuming perfect reciprocity and block fading, the received pilot signal is

$$\mathbf{S} = \sqrt{\rho}\,\mathbf{h}\mathbf{U} + \mathbf{n}, \tag{3}$$

where ρ is the average transmit power, $\mathbf{h} \in \mathbb{C}^{n_{\mathrm{rx}} \times n_{\mathrm{tx}}}$ is the channel to be estimated, and $\mathbf{n} \sim f_n(a, b)$ represents the general noise distribution as described in Section II-A. Each element of h can be written as $h_{ij} = \sqrt{L_{ij}}\, g_{ij}$, where L_{ij} captures the large-scale fading and $g_{ij} \sim \mathcal{CN}(0, 1)$ is the small-scale component. Spatial correlation is captured by the Kronecker model [40],

$$\mathbf{h} \;=\; \mathbf{R}_{\mathrm{rx}}^{1/2}\,\mathbf{h}_{\mathrm{iid}}\,\mathbf{R}_{\mathrm{tx}}^{1/2}, \quad \mathbf{h}_{\mathrm{iid}} \sim \mathcal{CN}\left(\mathbf{0}, \mathbf{I}\right), \tag{4}$$

where $\mathbf{R}_{\mathrm{rx}}$ and $\mathbf{R}_{\mathrm{tx}}$ denote the spatial correlation matrices at the receiver and transmitter sides, respectively. They model the antenna spatial correlation structures due to limited antenna spacing and propagation conditions. The channel estimator aims to minimize the normalized mean squared error (NMSE) between the estimated channel $\widehat{\mathbf{h}}$ and the true channel $\mathbf{h}$. The estimator $e(\cdot)$ considers the received pilot signal $\mathbf{S}$ and the pilot matrix $\mathbf{U}$ as inputs and outputs the estimated channel as:

$$\widehat{\mathbf{h}} = e(\mathbf{S}, \mathbf{U}). \tag{5}$$

If the noise term in (3) is neglected, the channel is retrieved by a right pseudoinverse of the pilot matrix, which forms the traditional LS estimator:

$$\widehat{\mathbf{h}}_{\mathrm{LS}} = \frac{1}{\sqrt{\rho}}\mathbf{S}\mathbf{U}^{\dagger}, \quad \mathbf{U}^{\dagger} = \mathbf{U}^H(\mathbf{U}\mathbf{U}^H)^{-1}, \tag{6}$$

where $\mathbf{U}^{\dagger}$ represents the Moore–Penrose pseudoinverse, which reduces to a simpler form due to the orthogonality condition imposed on the pilot matrix. This requires $\mathbf{U}$ to have at least as many columns as rows ($u \geq n_{\mathrm{tx}}$) and be full row rank. The LS estimator is unbiased and requires no prior channel statistics. However, under general noise conditions, its error covariance may significantly deviate from the idealized AWGN assumption and escalate rapidly under severe noise or impulsive disturbances.

In this proposed framework, the LS estimator is employed as a pre-estimating step, producing the noisy channel estimate $\widehat{\mathbf{h}}_{\mathrm{LS}}$. Subsequently, we utilize the proposed ZS-SS learning method to further improve the estimation accuracy of $\widehat{\mathbf{h}}_{\mathrm{LS}}$.

C. ZS-SS Learning Estimating Framework

Our proposed ZS-SS learning framework addresses the practical challenge that, in real-world deployments, only the pilot sequence is known. The ground-truth channel responses and prior knowledge of the noise distribution are unknown. Unlike traditional supervised or model-based approaches, our method leverages the intrinsic structure and redundancy of the observed data itself to guide network training. As a result, it does not require any label information or explicit knowledge of the noise model. Once trained in this manner, the ZS-SS

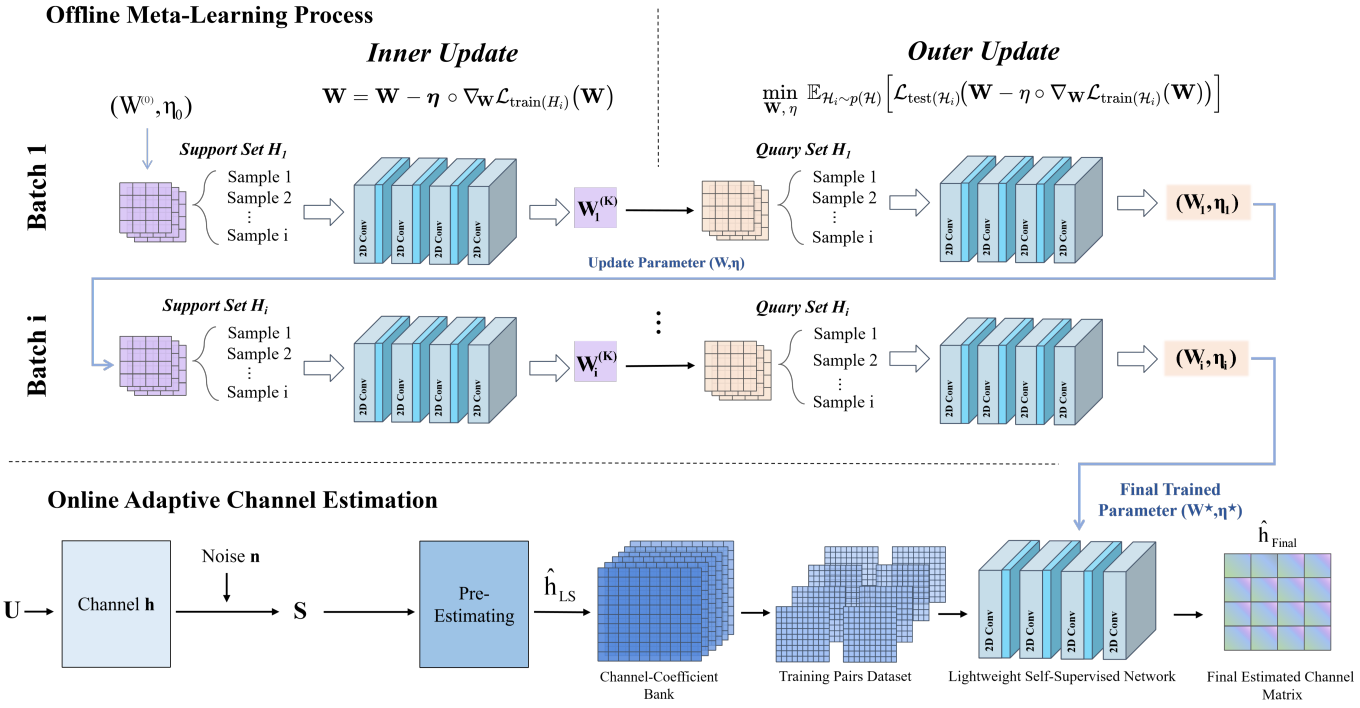

Fig. 1. The overall framework of the proposed ZS-SS learning method for channel estimation in massive MIMO systems.

learning estimator $f(\cdot;\boldsymbol{\theta})$ can be applied directly to $\widehat{\mathbf{h}}_{\mathrm{LS}}$ to produce a refined channel estimate:

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \ \mathrm{MSE}\left(f(\widehat{\mathbf{h}}_{\mathrm{LS}};\boldsymbol{\theta}),\ \mathbf{h}\right), \tag{7}$$

then

$$\hat{\mathbf{h}}_{\mathrm{Final}} = f\left(\widehat{\mathbf{h}}_{\mathrm{LS}};\boldsymbol{\theta}^*\right), \tag{8}$$

where $\boldsymbol{\theta}^*$ denotes the parameters learned from the self-supervised pairs. Furthermore, we employ Meta-SGD to pre-train the estimator, enabling online adaptation to new channel conditions as illustrated in Fig. 1. The meta-learning process is performed offline using historical channel data, while adaptation occurs online when a new channel realization is observed.

III. The Proposed ZS-SS Channel Estimation Method

In this section we first present the construction of a channel-coefficient bank that generates self-supervision pairs from a single noisy observation. We then detail the lightweight neural network architecture used to exploit this bank and conclude with the tailored loss functions design.

A. Construct Channel-Coefficient Bank and Pseudo Samples

The cornerstone of ZS-SS learning is to generate multiple training pairs from a single noisy observation. For the considered channel estimation task, the only measurement available online is the LS pre-estimate $\widehat{\mathbf{h}}_{\mathrm{LS}}$ obtained from (6). The typical approach for generating self-supervised training pairs is to use a downsampler, such as the diagonal downsampler [41]–[44]. However, this approach reduces data resolution and distorts noise characteristics, especially under non-Gaussian noise, leading to limited denoising performance. The main reason is that such downsampling alters the noise distribution and structure encountered during training.

To address these limitations, we design a more general and statistically robust approach by explicitly exploiting the non-local self-similarity inherent in wireless channel matrices. Non-local self-similarity describes the phenomenon where local coefficient patches in the channel matrix often have many similar non-local counterparts. This property has been extensively studied in image processing [45]–[47], and we observe that it also emerges in wireless communication channels. This arises from physical propagation effects such as scattering clusters, antenna array geometry, and slow spatial variations in multipath environments. By systematically identifying and mining these non-local but similar patches, we construct the channel-coefficient bank and training-pair dataset through the following two steps detailed below.

a) Channel-Coefficient Searching: To exploit non-local self-similarity of $\widehat{\mathbf{h}}_{\mathrm{LS}} \in \mathbb{C}^{n_{\mathrm{rx}} \times n_{\mathrm{tx}}}$, for every coefficient we extract a local $p \times p$ antenna patch

$$\mathbf{p}_{m,n} \ = \ \mathcal{P}_p\left(\widehat{\mathbf{h}}_{\mathrm{LS}}, m, n\right) \ \in \ \mathbb{C}^{p \times p}, \tag{9}$$

where $\mathcal{P}_p(\cdot)$ crops a square neighbourhood centred at (m,n). Within a larger search window of size $W \times W$ surrounding the same centre we collect the M non-overlapping patches whose distance to the reference patch is the smallest as shown in Fig. 2.

Remark 1. *An optimal patch size p exists for balancing noise robustness and matching accuracy. If p is too small, the extracted patch may fail to capture the true noise or channel features. Conversely, when p approaches W, it decreases the*

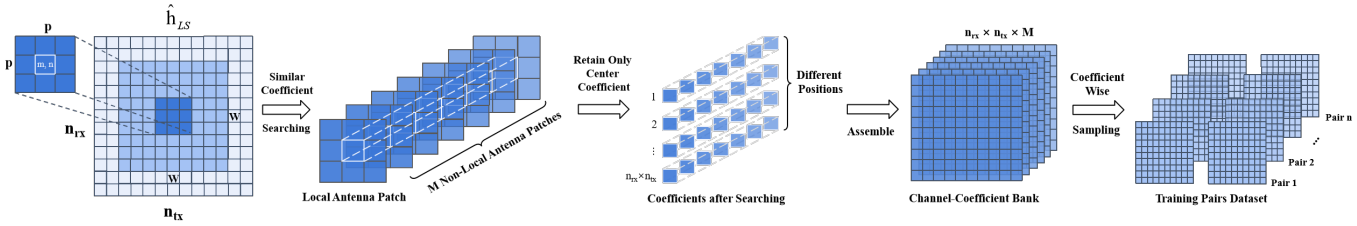

Fig. 2. Overview of constructing channel-coefficient bank and self-supervised training pairs dataset.

probability of finding genuinely similar patches due to overly restrictive matching conditions. Empirically, for $W = 22$, a range of p from 7 to 13 yields the empirically robust performance in our experiments.

Let (r, t) indexes antenna coordinates within the search window $\mathcal{W}_{m,n}$ centered at (m, n) and $\mathcal{I}_{m,n}$ be the indices of the M patches in $\mathcal{W}_{m,n}$ with the smallest distance to $\mathbf{p}_{m,n}$:

$$\mathcal{I}_{m,n} = \operatorname{argsort}_M \left\{ d(\mathbf{p}_{m,n},\, \mathcal{P}_p(\widehat{\mathbf{h}}_{\mathrm{LS}}, r, t)) : (r, t) \in \mathcal{W}_{m,n} \right\}. \tag{10}$$

Then,

$$\left\{ \mathbf{q}_{m,n}^{(i)} \right\}_{i=1}^{M} = \left\{ \mathcal{P}_p(\widehat{\mathbf{h}}_{\mathrm{LS}}, r, t) \mid (r, t) \in \mathcal{I}_{m,n} \right\}, \tag{11}$$

where $\operatorname{argsort}_M$ returns the indices of the M smallest elements, that is, $\operatorname{argsort}_M(x_1, \ldots, x_N) := (i_1, \ldots, i_M)$ such that $x_{i_1} \le \cdots \le x_{i_M}$ and $d(\cdot, \cdot)$ is the patch similarity metric of Frobenius norm.

Remark 2. *There exists an optimal value for M that balances sample diversity and similarity. When M is too small, the pseudo-sample pool is limited, which restricts network generalization. Conversely, as M increases, the similarity between selected patches and the reference degrades, reducing the quality of pseudo-training pairs.*

From each selected patch, we retain only its center coefficient as a pseudo-observation:

$$\tilde{h}_{m,n}^{(i)} = \left(\mathbf{q}_{m,n}^{(i)}\right)_{\lfloor p/2 \rfloor,\, \lfloor p/2 \rfloor}, \quad i = 1, \ldots, M, \tag{12}$$

where $\lfloor p/2 \rfloor$ denotes the center index of the selected $p \times p$ patch. Here, $\tilde{h}_{m,n}^{(i)}$ is the center coefficient of the i-th patch that is more similar to the reference patch $\mathbf{p}_{m,n}$, but is located at a distinct antenna position $(r_i, t_i) \in \mathcal{I}_{m,n}$ within the window.

In general, most $\tilde{h}_{m,n}^{(i)}$ come from different positions than (m, n), ensuring that the pseudo-observations are similar but not identical, and are corrupted by independent noise. This procedure ensures that the collected samples $\{\tilde{h}_{m,n}^{(i)}\}$ for each (m, n) are similar in their underlying channel structure, but distinct in their spatial origin and noise realization, thus providing multiple pseudo-observations for self-supervised learning.

Aggregating the M center coefficients collected for every antenna position (m, n) yields a third-order tensor, termed the channel-coefficient bank $\mathbf{B}$. Formally,

$$B_{m,n,i} \triangleq \tilde{h}_{m,n}^{(i)} = \left(\mathbf{q}_{m,n}^{(i)}\right)_{\lfloor p/2 \rfloor, \lfloor p/2 \rfloor}, \tag{13}$$

equivalently,

$$\mathbf{B} \triangleq \left(B_{m,n,i}\right)_{m=1,\ldots,n_{\mathrm{rx}};\; n=1,\ldots,n_{\mathrm{tx}};\; i=1,\ldots,M,} \tag{14}$$

where $\mathbf{B} \in \mathbb{C}^{n_{\mathrm{rx}} \times n_{\mathrm{tx}} \times M}$. Here, $\mathbf{B}$ stacks, for every antenna position (m, n), the M center coefficients extracted from the M most similar, non-overlapping $p \times p$ patches within the $W \times W$ window of $\widehat{\mathbf{h}}_{\mathrm{LS}}$. Thus $\{B_{m,n,i}\}_{i=1}^{M}$ are M pseudo-observations of the same underlying coefficient $h_{m,n}$ with approximately decorrelated noise as shown in the Phase I of Algorithm 1.

b) Channel-Coefficient Sampling: Once the channel-coefficient bank $\mathbf{B}$ has been assembled, training data are generated through coefficient-wise random sampling. For every antenna position (m, n) we draw two distinct indices $k_1, k_2 \in \{1, \ldots, M\}$ and form the noisy pair

$$\left(\hat{h}_{m,n}^{(k_1)}, \hat{h}_{m,n}^{(k_2)}\right), \quad k_1 \neq k_2, \tag{15}$$

where $\hat{h}_{m,n}^{(k)} = B_{m,n,k} = h_{m,n} + \varepsilon_{m,n}^{(k)}$ and $\varepsilon_{m,n}^{(k)}$ is the effective residual terms. Hence

$$\mathbb{E}\left[\hat{h}_{m,n}^{(k)}\right] = h_{m,n}, \quad \mathbb{E}\left[\varepsilon_{m,n}^{(k_1)} \varepsilon_{m,n}^{(k_2)*}\right] = 0. \tag{16}$$

Repeating (15) for all (m, n) yields the training pairs dataset for self-supervised learning

$$\mathcal{D} = \left\{ (\hat{h}_{m,n}^{(k_1)}, \hat{h}_{m,n}^{(k_2)}) \;\middle|\; (m, n) \in \Omega,\; 1 \le k_1 < k_2 \le M \right\}. \tag{17}$$

whose size is $|\mathcal{D}| = n_{\mathrm{rx}}\, n_{\mathrm{tx}} \binom{M}{2}$ and Ω denotes the set of all antenna positions (m, n) in the matrix. Coefficient-wise randomisation ensures that

$$\operatorname{Cov}\left(\varepsilon_{m,n}^{(k_1)}, \varepsilon_{m',n'}^{(k_2)}\right) \approx 0 \quad \text{for } (m, n) \neq (m', n'), \tag{18}$$

effectively suppressing the spatial correlation inherent to antenna elements. Random pairing thus supplies a quadratic number of statistically independent training pairs as shown in Fig. 2, mitigating over-fitting and strengthening robustness during online adaptation [48], [49]. This serves as implicit data augmentation [50], as even from a single noisy channel matrix, the network learns from a wide range of statistically diverse pseudo instances. With the channel-coefficient bank and self-supervised dataset effectively constructed, the next step is to design a compact yet powerful network to leverage this rich dataset.

B. Neural Network Architecture

To cope with the rapid variations of practical wireless channels and the tight real-time budget inherent to self-supervised operation, we design a lightweight four-layer fully convolutional CNN, as shown at the bottom of Fig. 1, that can be trained from scratch in a few hundred epochs. This four-layer

Algorithm 1: The Proposed ZS-SS Channel Estimation Method

Input: $\widehat{\mathbf{h}}_{\mathrm{LS}}$, patch size p, bank size M, window size W, number of epochs E, learning rate η.
Output: Final channel estimate $\hat{\mathbf{h}}_{\mathrm{Final}}$.

Phase I: Channel-Coefficient Bank Construction
for *each antenna position* (m, n) **do**
  Extract local patch: $\mathbf{p}_{m,n} = \mathcal{P}_p(\widehat{\mathbf{h}}_{\mathrm{LS}}, m, n)$.
  Find top-M similar patches in window $W \times W$ by (11).
  Retain center coefficients by (12).
  Then construct channel-coefficient bank by (13) and (14).
end
Phase II: Self-Supervised Network Training
for *epoch* $= 1$ **to** E **do**
  for *each antenna position* (m, n) **do**
    Randomly select indices $k_1, k_2 \in \{1, \ldots, M\}, k_1 \neq k_2$
    Form training pairs: $\big(\mathbf{B}[m, n, k_1], \mathbf{B}[m, n, k_2]\big)$.
  end
  Compute cross-reconstruction loss (20).
  Compute global clean estimate $\mathbf{h}_{\mathrm{Clean}} = f(\widehat{\mathbf{h}}_{\mathrm{LS}}; \boldsymbol{\theta})$.
  Construct clean coefficient bank $\widehat{\mathbf{B}}_{\mathrm{Clean}} = \mathrm{Bank}(\mathbf{h}_{\mathrm{Clean}})$.
  Compute sampling-consistency loss (23).
  Compute total loss (24).
end
Phase III: Channel Estimation Inference
Compute the final estimate: $\hat{\mathbf{h}}_{\mathrm{Final}} = f(\widehat{\mathbf{h}}_{\mathrm{LS}}; \boldsymbol{\theta}^*)$.
return $\hat{\mathbf{h}}_{\mathrm{Final}}$ `// Final channel estimate`

structure achieves a balance between expressiveness and efficiency, as deeper networks yield only marginal improvements but higher latency and overfitting, while shallower ones cannot capture essential channel correlations.

The real and imaginary parts of the LS pre-estimate are concatenated along the channel axis and fed to four 3×3 convolutions with 64 feature maps, each followed by a leaky ReLU with negative slope. A final 1×1 convolution maps the features back to the original channel dimension. Padding 1 is used for all 3×3 kernels, so the spatial resolution $n_{\mathrm{rx}} \times n_{\mathrm{tx}}$ is preserved. Residual connections, Batch Normalization (BN) and skip paths are omitted, as they tend to bias the network towards copying the noisy input and impede fast convergence. Weights are orthogonally initialised to stabilise gradients during the short training window.

Training is fully self-supervised, aiming to find the optimal parameter $\boldsymbol{\theta}^*$ in (7). At each optimisation step we randomly choose a set of antenna indices and, for every selected position, draw two distinct replicas from the channel-coefficient bank constructed earlier. One replica serves as the network input while the other acts as the pseudo target. The independence of the two noise realisations guarantees that the corresponding stochastic gradient is an unbiased estimate of the conditional-mean objective. After a few hundred epochs the network converges, and a single forward pass applied to $\widehat{\mathbf{h}}_{\mathrm{LS}}$ yields the final estimated channel matrix $\hat{\mathbf{h}}_{\mathrm{Final}}$.

C. Loss Function Design

Many self-supervised approaches such as DIP [14] and Self2Self [26] denoise by simply regressing the network output onto the same noisy observation,

$$\mathcal{L}(\boldsymbol{\theta}) = \Big\| f(Y_1; \boldsymbol{\theta}) - Y_2 \Big\|_2^2, \tag{19}$$

where Y_1 and Y_2 are two noisy realizations of the same underlying clean channel. Because the target still contains noise, this objective tends to overfit and requires delicate early stopping or additional regularisation, which may be not robust to complex noise, especially with non-Gaussian ones.

The proposed network is trained from scratch with a composite objective that contains a cross-reconstruction term $\mathcal{L}_{CR}$ and a sampling-consistency term $\mathcal{L}_{SC}$. During each epoch we first sample, for every antenna position (m, n), two statistically independent noisy replicas $Y^{(k_1)}$ and $Y^{(k_2)}$ from the channel-coefficient bank described in Section III-A.

a) Cross-Reconstruction Loss: The $\mathcal{L}_{CR}$ exploits the statistical independence of the two replicas. Each replica is passed through the network and compared with the other replica, so the error signal can only be reduced by recovering the clean coefficient that is common to both observations. Because the loss is symmetric in k_1 and k_2, it provides balanced gradients for every sample pair and prevents training bias:

$$\begin{aligned}\mathcal{L}_{CR}(\boldsymbol{\theta}) = \frac{1}{2}\Big(&\| f(Y^{(k_1)}; \boldsymbol{\theta}) - Y^{(k_2)} \|_2^2 + \\ &\| f(Y^{(k_2)}; \boldsymbol{\theta}) - Y^{(k_1)} \|_2^2 \Big).\end{aligned} \tag{20}$$

The symmetric structure of the $\mathcal{L}_{CR}$ in (20) leverages noise independence between pairs $(Y^{(k_1)}, Y^{(k_2)})$. Thus, minimizing this loss enforces the network to discard noise components $\varepsilon_{m,n}^{(k_1)}, \varepsilon_{m,n}^{(k_2)}$ and recover only their shared underlying clean signal $h_{m,n}$, ensuring unbiased gradients and preventing training bias.

b) Sampling-Consistency Loss: The $\mathcal{L}_{CR}$ alone assumes that the noisy bank perfectly matches the corruption process at inference time. However, repeated cropping and sampling introduce a slight distributional shift in practice. We compensate for this mismatch by adding a sampling-consistency regulariser that anchors every per-replica output to the denoised estimate of the full channel matrix. First, after the network parameters have been updated for the current mini-batch, we obtain a provisional clean estimate of the whole LS matrix $\mathbf{h}_{\mathrm{Clean}} = f\big(\widehat{\mathbf{h}}_{\mathrm{LS}}; \boldsymbol{\theta}\big)$. From $\mathbf{h}_{\mathrm{Clean}}$ we rebuild a clean version of channel-coefficient bank,

$$\widehat{\mathbf{B}}_{\mathrm{Clean}} = \mathrm{Bank}\big(\mathbf{h}_{\mathrm{Clean}}\big) \in \mathbb{C}^{n_{\mathrm{rx}} \times n_{\mathrm{tx}} \times M}, \tag{21}$$

by repeating the non-local search procedure of constructing the clean channel-coefficient bank. The surrogate clean coefficients paired with the current indices k_1 and k_2 are then

$$\widehat{X}^{(k_1)} = \widehat{\mathbf{B}}_{m,n,k_1}, \quad \widehat{X}^{(k_2)} = \widehat{\mathbf{B}}_{m,n,k_2}. \tag{22}$$

We encourage the network outputs for $Y^{(k_1)}$ and $Y^{(k_2)}$ to agree with these surrogates:

$$\begin{aligned}\mathcal{L}_{SC}(\boldsymbol{\theta}) = \frac{1}{2}\Big(&\| f(Y^{(k_1)}; \boldsymbol{\theta}) - \widehat{X}^{(k_1)} \|_2^2 + \\ &\| f(Y^{(k_2)}; \boldsymbol{\theta}) - \widehat{X}^{(k_2)} \|_2^2 \Big).\end{aligned} \tag{23}$$

The $\mathcal{L}_{SC}$ in (23) anchors the network outputs to the global denoised bank $\widehat{\mathbf{B}}_{\text{Clean}}$, mitigating the distributional shift introduced by random sampling even when minor discrepancies arise between training and inference distributions. This loss can be viewed intuitively as a self-consistency regularization. It encourages the network's output for sampled noisy inputs to remain close to its prediction on the full noisy observation, thus stabilizing training even when the sampling introduces minor statistical deviations [41], [51]. The two objectives are summed with equal weight,

$$\mathcal{L}_{total}(\boldsymbol{\theta}) = \mathcal{L}_{CR}(\boldsymbol{\theta}) + \mathcal{L}_{SC}(\boldsymbol{\theta}), \tag{24}$$

so that the network simultaneously learns to cross-predict between noisy pairs and to remain consistent with its self-denoised reference as shown in the Phase II of Algorithm 1. See **Appendix A** for theoretical analysis of this loss design.

IV. Adaptability Enhancement via Meta-Learning

In this section, we introduce the Meta-SGD algorithm to equip our proposed ZS-SS estimator with fast inference time and strong generalisation capabilities across varying channel scenarios.

A. The Meta-Learning Framework

Although the ZS-SS learning scheme in Section III needs no labels, it typically requires hundreds of gradient steps and is sensitive to SNR and propagation conditions. Fixed learning-rate schedules can be either slow or overfit, and heuristic early stopping is unreliable across scenarios. To achieve fast inference time yet stable adaptation we embed the proposed ZS-SS learning method in the meta-learning framework Meta-SGD [52] as shown in Fig. 1. We treat each pre-collected LS-estimated channel dataset, denoted as a task instance H, as detailed in Section IV-B. These tasks are sampled from a distribution $p(H)$. Offline meta-training draws a diverse task set $\mathcal{H} = \{H_i\}$ that covers varying antenna layouts, SNRs, and channel conditions and splits each H_i into a support set $\text{train}(H_i)$ and a query set $\text{test}(H_i)$. The meta-learner maintains an initial parameter vector $\mathbf{W}$ for $f(\cdot;\mathbf{W})$ and an element-wise step-size vector $\boldsymbol{\eta} \in \mathbb{R}^{\dim(\mathbf{W})}$. For each task, the inner loop performs K gradient steps on $\text{train}(H_i)$ using $\boldsymbol{\eta}$ and the outer loop updates both $\mathbf{W}$ and $\boldsymbol{\eta}$ to reduce the loss on $\text{test}(H_i)$ averaged over tasks. Specifically, as illustrated in Fig. 1, $\mathbf{W}^{(0)}$ denotes the initial meta-parameters, and $\mathbf{W}_1$ represents the updated parameters after the first optimization. Iterating yields a meta-initialisation $(\mathbf{W}^\star, \boldsymbol{\eta}^\star)$ that encodes broad channel priors and an element-wise learning-rate schedule suited to our self-supervised objective. In the online phase, the freshly received LS matrix is regarded as a new task. Starting from $\mathbf{W}^\star$, a few gradient steps with $\boldsymbol{\eta}^\star$ produce adapted weights $\mathbf{W}^{\star'}$, enabling near-peak accuracy within a few epochs without retraining from scratch. The detailed inner/outer updates and the offline/online procedures are given in Sections IV-C and IV-D.

Algorithm 2: The Offline Meta-Training Process of the Proposed Method

Input: Meta-training dataset $\mathcal{D}_{\text{meta}} = \{H_i\}_{i=1}^N$, number of meta-training iterations T_{outer}, number of inner-loop steps K, learning rates α, α_η, initial neural network architecture and hyperparameters.
Output: Meta-learned neural network parameters $\mathbf{W}^\star$ and element-wise adaptive step-sizes $\boldsymbol{\eta}^\star$.

Initialize $\mathbf{W}, \boldsymbol{\eta}$ randomly.
for *iteration* $= 1$ **to** T_{outer} **do** // Outer loop
  Sample a mini-batch of tasks $\{H_i\}_{i=1}^M$ from $\mathcal{D}_{\text{meta}}$.
  foreach H_i *in mini-batch* **do** // Inner loop
    Randomly split H_i into support set $\text{train}(H_i)$ and query set $\text{test}(H_i)$.
    Set adapted parameters: $\mathbf{W}_i^{(0)} \leftarrow \mathbf{W}$.
    for $t = 0$ **to** $K - 1$ **do**
      Compute inner gradient on support set by (25).
    end
    Compute task-specific loss on query set: $\ell_i \leftarrow \mathcal{L}_{\text{test}(H_i)}(\mathbf{W}_i^{(K)})$.
  end
  Compute meta-objective over mini-batch: $\mathcal{J}(\mathbf{W}, \boldsymbol{\eta}) \leftarrow \frac{1}{M}\sum_{i=1}^M \ell_i$.
  Compute meta-gradients with respect to $(\mathbf{W}, \boldsymbol{\eta})$ by back-propagation through inner updates.
  Update meta-parameters with outer-loop gradient descent by (28).
end
return Optimized meta-parameters $(\mathbf{W}^\star, \boldsymbol{\eta}^\star)$.

B. The Meta-Learning Dataset

In practical deployments a massive number of LS-estimated channel matrices can be collected almost for free since every uplink pilot transmission yields a noisy channel realization sample, denoted by H. Although the noise level, spatial correlation and propagation geometry vary from frame to frame, these matrices also share structural traits such as block fading and reciprocity. Our goal is to provide the proposed framework with a well-informed starting point that remains agnostic to hand-crafted priors, so it can accommodate a wide range of noise statistics without having to learn from scratch each time.

With this aim we compile a meta-training set $\mathcal{D}_{\text{meta}} = \{H_i\}_{i=1}^N$, containing only raw LS outputs gathered under different SNRs, antenna configurations, Doppler spreads and LoS/NLoS conditions, with no clean channel labels required. Because the samples arise naturally during normal operation, assembling $\mathcal{D}_{\text{meta}}$ imposes negligible overhead and keeps the overall complexity unchanged. For every task H_i we randomly assign a subset of its antenna positions to a support set and the remainder to a query set. The support set drives the inner adaptation of Meta-SGD, while the query set evaluates how well the adapted parameters generalise to unseen coefficients within the same channel realisation.

This label-free, diversity-driven design endows the meta-learner with rich prior knowledge of realistic noise patterns and channel structures, ensuring that the subsequent online adaptation phase benefits from a broad and representative foundation. This dataset feeds the meta-training process detailed next.

C. Offline Meta-Training Process

Meta-training operates on the label-free collection $\mathcal{D}_{\text{meta}} = \{H_i\}_{i=1}^{N}$ introduced in Section IV-B. Each H_i is turned into a task by randomly partitioning its antenna positions into a support set $\text{train}(H_i)$ and a query set $\text{test}(H_i)$.

a) Inner update: Starting from the shared parameters $\mathbf{W}$ and the element-wise step-size vector $\boldsymbol{\eta}$, the estimator performs K gradient steps on $\text{train}(H_i)$:

$$\begin{aligned} \mathbf{W}^{(0)} &= \mathbf{W}, \\ \mathbf{W}^{(t+1)} &= \mathbf{W}^{(t)} - \boldsymbol{\eta} \circ \nabla_{\mathbf{W}^{(t)}} \mathcal{L}_{\text{train}(H_i)}\big(\mathbf{W}^{(t)}\big) \end{aligned} \tag{25}$$

for $t = 0, 1, \ldots, K-1$, where $\circ$ denotes element-wise multiplication. The resulting adapted parameters $\mathbf{W}_i^{(K)}$ are task-specific and obtained entirely from noisy inputs, without requiring any clean channel labels. Here, the training loss used in the inner loop shares the self-supervised composite form defined earlier in Section III-C. The test loss $\mathcal{L}_{\text{test}(H_i)}(\mathbf{W})$ employed in the outer loop has the same formulation but is evaluated exclusively on the query set.

b) Outer update: The meta-objective evaluates how well the adapted weights generalise to the query coefficients:

$$\mathcal{J}(\mathbf{W}, \boldsymbol{\eta}) = \mathbb{E}_{H_i \sim p(\mathcal{H})}\Big[\mathcal{L}_{\text{test}(H_i)}\big(\mathbf{W}_i^{(K)}\big)\Big]. \tag{26}$$

In practice, we approximate this expectation by averaging over a mini-batch of M tasks randomly sampled from $\mathcal{H}$:

$$\mathcal{J}(\mathbf{W}, \boldsymbol{\eta}) \approx \frac{1}{M} \sum_{i=1}^{M} \mathcal{L}_{\text{test}(H_i)}\big(\mathbf{W}_i^{(K)}\big). \tag{27}$$

Back-propagation through the K unrolled inner steps yields the gradients of $\mathcal{J}$ with respect to both $\mathbf{W}$ and $\boldsymbol{\eta}$, which are used by a standard optimiser to update the meta-parameters as follows:

$$\mathbf{W} \leftarrow \mathbf{W} - \alpha \nabla_{\mathbf{W}} \mathcal{J}(\mathbf{W}, \boldsymbol{\eta}), \quad \boldsymbol{\eta} \leftarrow \boldsymbol{\eta} - \alpha_\eta \nabla_{\boldsymbol{\eta}} \mathcal{J}(\mathbf{W}, \boldsymbol{\eta}), \tag{28}$$

where α and α_η denote the learning rates of the outer optimizer.

Remark 3. *For non-Gaussian or time-varying noise, a small K with a large T_{outer} is preferable, enabling the model to learn a robust initialisation from diverse tasks rather than over-refining on individual tasks. A small inner-loop step K yields fast memory-efficient updates and promotes better cross-task generalisation. In contrast, a large K improves per-task fitting but increases the risk of over-fitting support-set noise.*

Repeating the inner-outer cycle over a large, SNR-diverse mini-batch of tasks for T_{outer} iterations produces the meta-initialisation $(\mathbf{W}^\star, \boldsymbol{\eta}^\star)$ as shown in Algorithm 2 and the corresponding part in Fig. 1, which is then passed to the online adaptation stage in Section IV-D.

D. Online Meta-Adaption Process

In online deployment, the estimator must adapt to each newly received LS-estimated channel realization. Given the offline-trained meta-initialisation $(\mathbf{W}^\star, \boldsymbol{\eta}^\star)$, the online adaptation is both algorithmically fast at inference and straightforward: upon observing a new noisy channel matrix $\widehat{\mathbf{h}}_{\text{LS}}$, the model performs only a few gradient-based adaptation steps starting from $\mathbf{W}^\star$, using the pre-calibrated learning rates $\boldsymbol{\eta}^\star$. Specifically, we first construct the channel-coefficient bank from the observed channel realization as described in Section III-A. Each online adaptation step then minimizes the self-supervised loss function, computed exclusively from randomly selected pairs of noisy coefficient samples drawn from the channel-coefficient bank. Given the meta-initialisation, the parameters are updated rapidly as follows:

$$\mathbf{W}^{(t+1)} = \mathbf{W}^{(t)} - \boldsymbol{\eta}^\star \circ \nabla_{\mathbf{W}^{(t)}} \mathcal{L}_{\text{online}}(\mathbf{W}^{(t)}) \tag{29}$$

for $t = 0, 1, \ldots, K_{\text{online}} - 1$, where typically $K_{\text{online}} \ll K$. After these few gradient steps, we directly obtain the adapted estimator parameters $\mathbf{W}^{\star'}$. Inference then proceeds immediately by applying the adapted estimator to the full noisy channel:

$$\hat{\mathbf{h}}_{\text{Final}} = f(\widehat{\mathbf{h}}_{\text{LS}}; \mathbf{W}^{\star'}). \tag{30}$$

Moreover, the online meta-adaptation supports continuous parameter refinement when processing a sequence of multiple incoming channel matrices. Consider a sequence of LS-estimated channel matrices $\widehat{\mathbf{h}}_{\text{LS}}^{(1)}, \widehat{\mathbf{h}}_{\text{LS}}^{(2)}, \ldots, \widehat{\mathbf{h}}_{\text{LS}}^{(N)}$, the model can continuously adapt its parameters to dynamically track the evolving channel distribution. Concretely, starting from the meta-initialisation parameters $\mathbf{W}^\star$, the adaptation parameters after processing each channel realization are recursively updated:

$$\begin{aligned} \mathbf{W}^{(0)} &= \mathbf{W}^\star, \\ \mathbf{W}^{(n)} &= \mathbf{W}^{(n-1)} - \boldsymbol{\eta}^\star \circ \nabla_{\mathbf{W}^{(n-1)}} \mathcal{L}_{\text{online}}\big(\mathbf{W}^{(n-1)}; \widehat{\mathbf{h}}_{\text{LS}}^{(n)}\big), \end{aligned} \tag{31}$$

$n = 1, \ldots, N$, where each channel realization $\widehat{\mathbf{h}}_{\text{LS}}^{(n)}$ has its own constructed channel-coefficient bank for computing the self-supervised loss. Subsequently, inference for the n-th channel is conducted with:

$$\hat{\mathbf{h}}_{\text{Final}}^{(n)} = f(\widehat{\mathbf{h}}_{\text{LS}}^{(n)}; \mathbf{W}^{(n)}). \tag{32}$$

Algorithm 3: The Online Meta-Adaptation Process of the Proposed Method

Input: Offline-trained meta-parameters $(\mathbf{W}^\star, \boldsymbol{\eta}^\star)$, incoming LS-estimated channel realizations $\{\widehat{\mathbf{h}}_{\text{LS}}^{(n)}\}_{n=1}^{N}$ observed sequentially.

Output: Final channel estimates $\{\hat{\mathbf{h}}_{\text{Final}}^{(n)}\}_{n=1}^{N}$ corresponding to each incoming noisy channel realization.

Online Adaptation and Inference Loop:
Set $\mathbf{W}^{(0)} \leftarrow \mathbf{W}^\star$
for $n = 1$ **to** N **do**
  Observe new online channel realization $\widehat{\mathbf{h}}_{\text{LS}}^{(n)}$.
  `// Perform online adaptation:`
  for $t = 0$ **to** $K_{online} - 1$ **do**
    Compute online gradient step by (29).
  end
  Set adapted parameters: $\mathbf{W}^{(n)} \leftarrow \mathbf{W}^{(K_{\text{online}})}$
  `// Inference using adapted parameters:`
  Compute final channel estimate:
  $\hat{\mathbf{h}}_{\text{Final}}^{(n)} \leftarrow f(\widehat{\mathbf{h}}_{\text{LS}}^{(n)}; \mathbf{W}^{(n)})$
  `// Continuous tracking:`
  $\mathbf{W}^{(0)} \leftarrow \mathbf{W}^{(n)}$
end
return Final denoised channel estimates $\{\hat{\mathbf{h}}_{\text{Final}}^{(n)}\}_{n=1}^{N}$

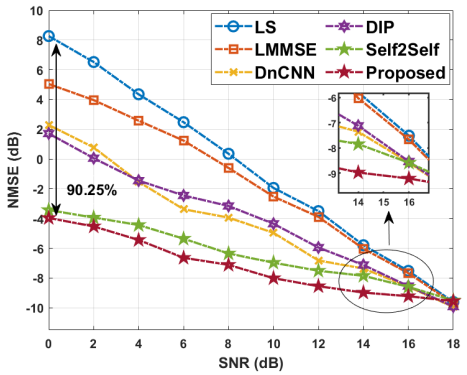

Fig. 3. The NMSE versus the SNR in Gaussian noise ($a = 1, b = 0$).

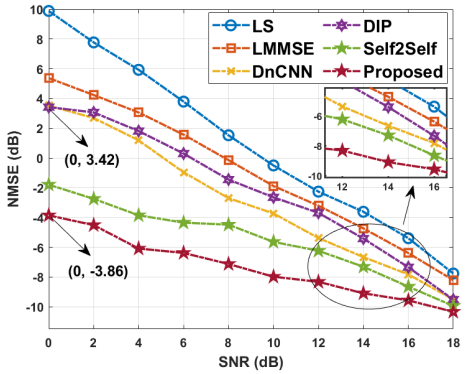

Fig. 4. The NMSE versus the SNR in Laplacian noise ($a = 0, b = 1$).

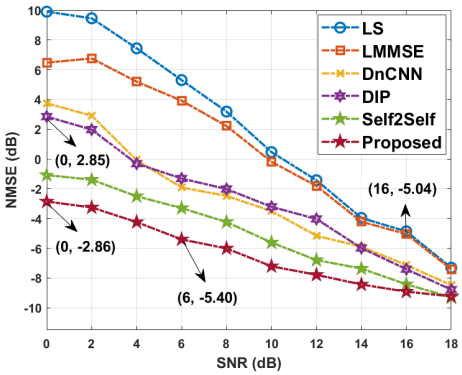

Fig. 5. The NMSE versus the SNR in BGIN ($a = 0, b = 1$).

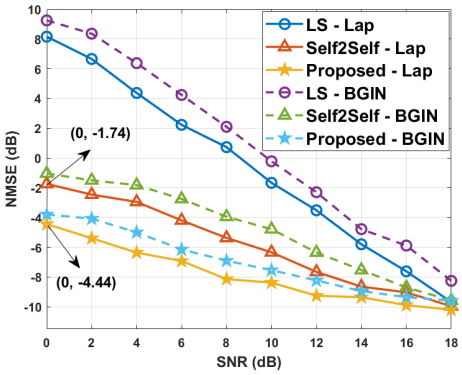

Fig. 6. The NMSE versus the SNR in noise mixture scenario ($a = 0.5, b = 0.5$).

This online adaptation mechanism leverages Meta-SGD, allowing model parameters to quickly track dynamically changing channel conditions and improve adaptability, as detailed in Algorithm 3. Such capability is particularly advantageous in wireless systems with evolving channel statistics caused by mobility or environmental changes. The entire adaptation-inference loop is lightweight and enables rapid estimation, thus avoiding the excessive overhead and latency of repeated conventional retraining in dynamic scenarios.

V. Simulation Results

In this section, we numerically validate the effectiveness of the proposed algorithm by evaluating its NMSE performance, defined explicitly as:

$$\text{NMSE} = \mathbb{E}\left[\frac{\|\hat{\mathbf{h}}_{\text{Final}} - \mathbf{h}\|_F^2}{\|\mathbf{h}\|_F^2}\right], \tag{33}$$

where $\mathbb{E}[\cdot]$ denotes the expectation and $\|\cdot\|_F$ represents the Frobenius norm. Unless otherwise stated, the simulation results are presented in dB scale.

A. Experimental Details

All channel realizations employed in the simulations are generated following the 3GPP channel model within the QuaDRiGa simulation framework [53], executed in the MATLAB. Unless otherwise stated, the scenario is set to the 3GPP 38.901 Urban Micro (UMi) LoS model, and the default system parameters are configured as follows: the number of receive antennas $n_{\text{rx}} = 64$, the number of transmit antennas $n_{\text{tx}} = 32$, and the pilot sequence length is set to 36. Our deep learning-based channel estimation method is implemented using PyTorch 2.0.1 with CUDA 11.7, and executed on a computational platform equipped with an NVIDIA RTX 3090 GPU with 24 GB memory, alongside a 20-core AMD EPYC 7642 CPU.

We summarize the experimental parameters and hyperparameters settings as follows. For channel-coefficient bank construction, we set the patch size for local coefficient extraction as $p = 9$, non-local similarity search window $W = 22$, $M = 10$ nearest neighbors for constructing the channel-coefficient bank, and sample 5 slices per training batch. To mitigate excessive GPU memory consumption, the block size for partial matrix processing is configured as 16. Model training runs with an initial learning rate of 1×10^{-4}. In Meta-SGD, the outer-loop learning rate is 1×10^{-3} over 400 iterations, each with 10 inner-loop steps, and tasks are split with a support ratio of 0.8.

To comprehensively assess the performance of our proposed method, we consider a range of baseline algorithms spanning both traditional methods and state-of-the-art deep learning-based approaches, including supervised and self-supervised paradigms.

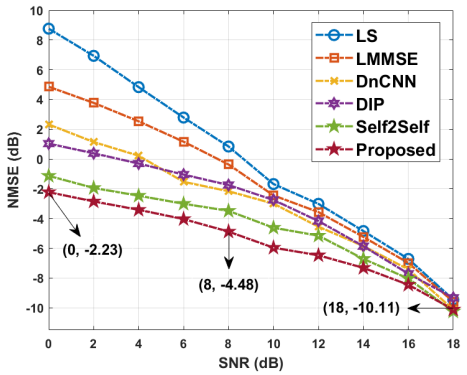

Fig. 7. The NMSE versus the SNR in NLoS.

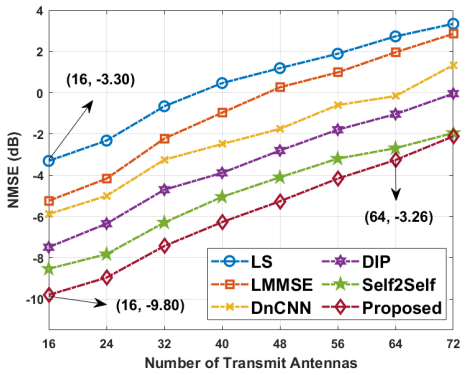

Fig. 8. The NMSE versus the number of transmit antennas.

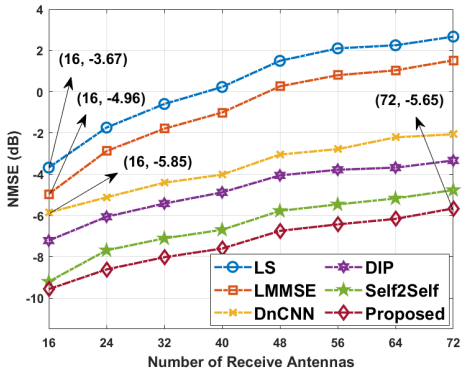

Fig. 9. The NMSE versus the number of receive antennas.

For traditional benchmarks, we employ the traditional LS and LMMSE estimators. For the supervised deep learning baseline, we utilize DnCNN [24], [25], a popular model that requires extensive labeled datasets and prolonged offline training. Following the original methodology in [24], we train two distinct DnCNN models separately at low and high SNRs (specifically at 6 dB and 12 dB, respectively) and subsequently apply the trained models across their corresponding SNR regimes, as training and maintaining a dedicated model for every single SNR level is impractical for real-world deployment. We train the model for most 500 iterations, utilizing a batch size of 128 and an initial learning rate of 10^{-3}. The training, testing and validation sets consist of 32000, 4000 and 4000 channel samples, respectively. In the self-supervised category, we select DIP [14] and the recently proposed Self2Self [26] algorithm. Notably, Self2Self currently represents the state-of-the-art self-supervised approach for this task, although it necessitates a large-scale neural network structure and significant computational time during inference. Consistent with the implementation in [26], [36], we employ a 5-level U-Net with partial convolutions and optimize the network for a fixed budget of 5000 iterations per instance (without heuristic early-stopping) with a learning rate of 10^{-4}.

We evaluate performance under both Gaussian and non-Gaussian noise to comprehensively assess the robustness and generalization capability of our proposed method. The non-Gaussian cases follow widely used models in the literature. The complex Laplacian noise is generated as in [30], by combining independent real and imaginary components from a Laplace distribution, with scale parameter $b = \sqrt{\sigma^2}/2$ to match the variance of the Gaussian baseline. The BGIN model follows [37], where the received noise is $\mathbf{n}_{\mathrm{BG}} = \mathbf{n}_{\mathrm{w}} + \beta\mathbf{n}_{\mathrm{i}}$, with $\beta \in \{0, 1\}$ an i.i.d. Bernoulli switch of probability 0.01. Thermal noise variance is $\sigma_w^2 = E_s n_{\mathrm{tx}}/\mathrm{SNR}$ and impulsive noise variance is $\sigma_i^2 = \bar{K}\sigma_w^2$ with $\bar{K} = 18\,\mathrm{dB}$. The Laplacian noise scenario serves as the default baseline throughout our experiments. All noise processes are i.i.d. across channel coefficients, and the non-Gaussian scenarios are configured to have identical average noise power as the Gaussian baseline for fair comparison.

B. Numerical Results

Fig. 3 presents NMSE versus SNR under Gaussian noise ($a = 1, b = 0$) for all compared methods. Our approach consistently yields the lowest estimation error, substantially outperforming traditional LS and LMMSE estimators across the SNR range, with error reductions exceeding 90% at low-SNR regime. The supervised DnCNN method achieves strong results at 6 and 12 dB due to SNR-specific training, but elsewhere performs similarly to the self-supervised DIP method. Self2Self surpasses other baselines, yet our method retains a clear advantage, particularly at moderate and high SNRs, confirming its robustness and overall effectiveness.

Fig. 4 shows the NMSE performance under Laplacian noise ($a = 0, b = 1$). All baselines suffer noticeable degradation in non-Gaussian conditions, with LS and LMMSE experiencing significant declines. Despite this, our method maintains robust superiority, outperforming all other approaches by a considerable margin. At low SNR, our algorithm improves estimation accuracy by over five times relative to both the supervised DnCNN and the self-supervised DIP methods. Even the advanced Self2Self baseline is notably challenged in the presence of non-Gaussian noise, while our method achieves substantial gains, underscoring its enhanced robustness and adaptability to diverse noise environments.

Fig. 5 presents the NMSE results under BGIN ($a = 0, b = 1$). The presence of significant impulsive noise degrades the performance of all methods noticeably, including ours. Tradi-

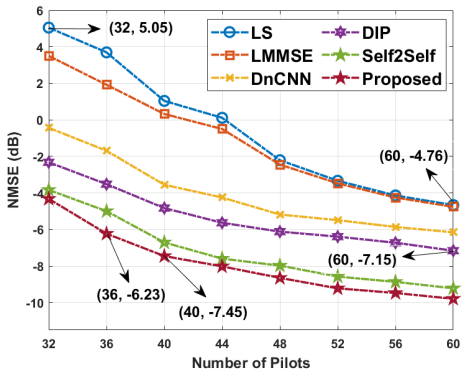

Fig. 10. The NMSE versus the pilot length.

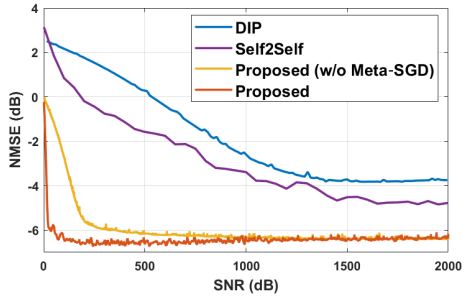

Fig. 11. The NMSE versus the epoch.

tional estimators are particularly impacted, further widening the gap with deep learning-based approaches. DnCNN and DIP remain comparably effective despite the impulsive interference. Notably, our proposed method consistently outperforms Self2Self and other baselines, demonstrating significant robustness and superior capability for mitigating the detrimental impact of impulsive interference in realistic wireless communication environments.

Fig. 6 presents the NMSE results under the mixed-noise setting ($a = 0.5, b = 0.5$), where the non-Gaussian component is either Laplacian or BGIN. The composite interference elevates error for all estimators, with the BGIN mixture proving more detrimental than the Laplacian mixture. While Self2Self improves upon LS across SNRs, our proposed method consistently achieves the lowest NMSE for both mixtures, with particularly clear margins at low–to–moderate SNR. These results confirm the distribution-agnostic design of our estimator and its robustness when Gaussian and non-Gaussian disturbances coexist.

Fig. 7 presents the channel estimation performance evaluated in terms of NMSE under a NLoS scenario. Compared to LoS conditions, the performance differences among the various algorithms become noticeably more compact across the entire SNR range. This phenomenon can be attributed to the inherent complexity and randomness of NLoS channels, characterized by severe multipath fading and lack of dominant direct paths, which collectively pose greater challenges for accurate channel estimation. Nevertheless, our proposed method consistently achieves the best or near-best estimation performance, notably outperforming traditional approaches such as LS by more than an order of magnitude at low SNR regimes. These results further demonstrate the efficacy and robustness of our approach in challenging NLoS propagation environments.

Figs. 8 - 9 show the NMSE performance as the number of transmit and receive antennas increases from 16 to 72. With more antennas, the number of coefficients grows and the per-antenna pilot allocation decreases, making estimation increasingly difficult and resulting in higher errors for all methods. Despite this, our proposed algorithm consistently achieves the lowest NMSE across the varying antenna configurations. Notably, our method with 64 transmit antennas achieves comparable performance with the traditional LS baseline at 16 antennas, and with 72 receive antennas rivals DnCNN at 16, clearly outperforming traditional approaches.

Fig. 10 depicts the impact of varying the number of pilot symbols on the channel estimation performance for different methods. It is clear that our proposed approach consistently achieves the lowest NMSE across the entire range of pilot numbers. Notably, even under severely limited pilot resources, such as with only 36 pilots, our method attains superior estimation accuracy, outperforming traditional approaches that employ 60 pilots. These results highlight the outstanding data efficiency and robustness of our approach in scenarios with scarce pilot resources.

C. Complexity & Convergence Analysis

Table I compares the computational resources of all methods, including run-time, model size, and memory usage. Our approach achieves excellent efficiency, with GPU and CPU run-times only 1.48% and 0.26% of Self2Self, respectively, thanks to its lightweight network and meta-learning strategy that significantly accelerates convergence during inference. It is worth noting that these results are based on unoptimized FP32 implementation. In practical deployment, further acceleration can be achieved via dedicated inference engines (e.g., TensorRT) and model quantization (e.g., INT8/INT4) on NPUs or FPGAs, which typically offer multi-fold speedups [54]. Because DnCNN is trained offline with significant amounts of labeled data, its inference time at deployment is almost negligible. Each forward pass requires just 230 M FLOPs, much less than Self2Self (990 M) and DnCNN (1366 M), and meta-learning adds no extra FLOPs during inference. The total proposed model size is only 225 k including meta-parameters, about ten times smaller than DIP and much less than other deep learning baselines. Memory usage is also modest at 130 MB, nearly ten times less than self-supervised competitors. These characteristics make our method highly suitable for low-latency and resource-constrained wireless communication applications, without compromising estimation performance.

Fig. 11 shows that while Self2Self and DIP need about 1500 epochs to converge, our method achieves lower NMSE in just 350 epochs without meta-learning, and only 150 epochs with meta-learning enabled. This demonstrates much faster convergence and higher accuracy, underscoring the efficiency

TABLE I
COMPARISON OF COMPUTATIONAL RESOURCES AMONG DIFFERENT METHODS.

Method	Proposed	Proposed (w/o Meta-SGD)	DIP	Self2Self	DnCNN
GPU Run Time (sec.)	2.82	6.17	58.91	190.13	–
Percentage	1.48%	3.23%	30.98%	100%	–
CPU Run Time (sec.)	3.34	6.31	107.48	1279.34	–
Percentage	0.26%	0.49%	8.40%	100%	–
Model Parameters (k)	224.51	113.13	2230.00	1120.00	669.38
FLOPs (M)	229.11	229.11	580.82	990.25	1365.61
Memory (MB)	131.40	129.80	972.90	4720.41	3912.00

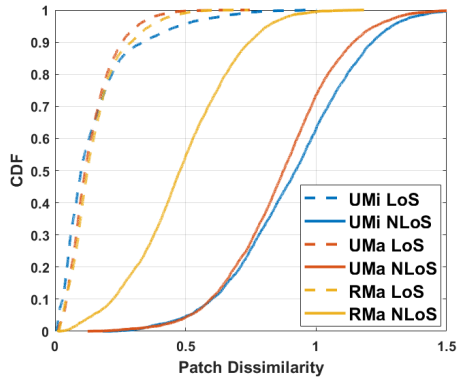

Fig. 12. CDF of the average patch matching error across diverse propagation scenarios.

and practical advantages of our meta-learned self-supervised framework for robust channel estimation.

D. Generalization and Robustness Analysis

1) Robustness of Non-Local Self-Similarity: To further validate the generalization capability, we analyze the stability of the core non-local self-similarity assumption across UMi, Urban Macro (UMa), and Rural Macro (RMa) environments. As shown in Fig. 12, while LoS scenarios naturally exhibit the highest similarity, the patch matching error in NLoS conditions remains bounded significantly below that of uncorrelated patterns, despite severe multipath scattering. These results demonstrate that the proposed feature extraction mechanism is statistically robust against diverse propagation scenarios. Even in the most complex urban environments, the non-local self-similarity property is preserved sufficiently to distinguish channel features from noise.

2) Performance under α-Stable Noise: To further assess the robustness against heavy-tailed interference, we conducted experiments using the α-Stable noise model. Table II presents the NMSE performance at 10 dB with varying characteristic exponents α. Traditional estimators and the supervised baseline exhibit severe performance degradation under impulsive conditions. In contrast, the proposed ZS-SS framework maintains robust performance across varying characteristic exponents, exhibiting strong resilience to impulsive outliers while preserving high accuracy in the Gaussian regime.

TABLE II
NMSE (dB) COMPARISON UNDER α-STABLE NOISE.

Method	$\alpha = 1.4$	$\alpha = 1.6$	$\alpha = 1.8$	$\alpha = 2.0$
LS	16.50	10.82	7.05	5.13
LMMSE	12.09	9.29	3.91	1.56
DnCNN	8.22	2.58	-0.94	-2.81
Self2Self	1.18	-0.60	-2.70	-2.84
Proposed	**-0.09**	**-0.80**	**-4.06**	**-4.86**

TABLE III
ROBUSTNESS AGAINST SPATIALLY COLORED NOISE

Noise Type	Correlation (ρ)	NMSE (dB)
Independent (Baseline)	0.0	-8.29
Weak Correlation	0.2	-7.98
Moderate Correlation	0.5	-7.80
Strong Correlation	0.8	-7.42

3) Robustness to Spatially Colored Noise: We further evaluated the robustness of the proposed ZS-SS framework under spatially colored noise at SNR $= 10$ dB, modeled via an exponential correlation matrix with coefficient ρ. As shown in Table III, while NMSE naturally degrades as ρ increases, the method remains robust even under strong correlation ($\rho = 0.8$), which is attributed to the decorrelation capability of non-local search and the global regularization imposed by the $\mathcal{L}_{sc}$.

4) Performance of Spectral Efficiency in System-Level: To validate the practical impact of the improved estimation accuracy, we evaluated the achievable spectral efficiency in a downlink MIMO system using zero-forcing precoding. As shown in Fig. 13, the LS baseline suffers from noise amplification, resulting in suboptimal precoding vectors and limited capacity. In contrast, the proposed method demonstrates significant robustness across the SNR range. This confirms that the proposed estimator effectively prevents precoding distortion, leading to substantial system-level throughput gains.

E. Ablation Study

1) Impact of the Patch Size p and the Non-Overlapping Patches Size M: Fig. 14 reports NMSE versus the patch size p and the non-overlapping patches size M. For p, the NMSE drops quickly from $p = 3$ to $p \in [7, 9]$, attains its minimum

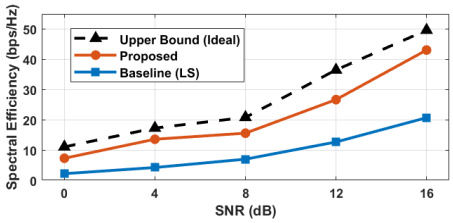

Fig. 13. The spectral efficiency (bps/Hz) versus the SNR.

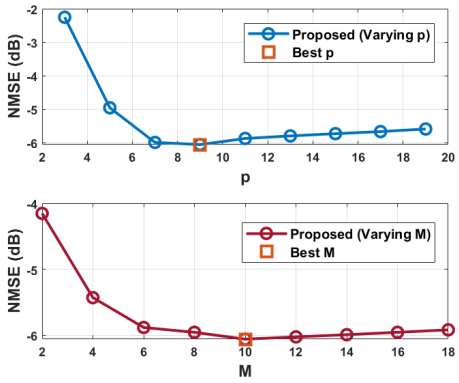

Fig. 14. The NMSE versus different p and M.

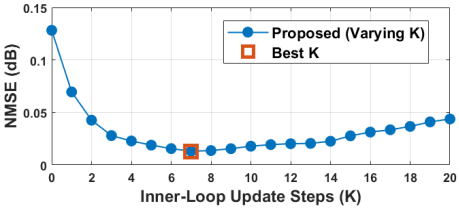

Fig. 15. The NMSE versus inner-loop update steps (K) under non-Gaussian noise.

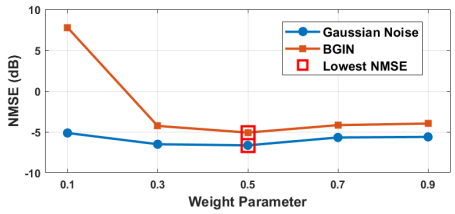

Fig. 16. Sensitivity analysis of the loss weight parameter under Gaussian and BGIN scenarios.

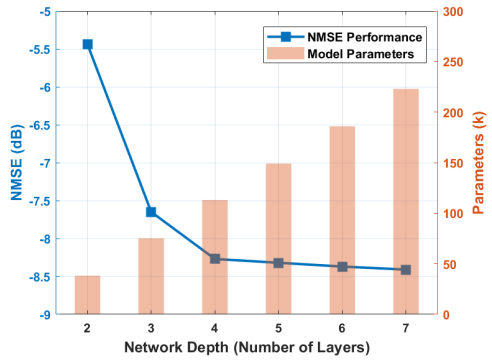

Fig. 17. Trade-off analysis between network depth and NMSE performance.

TABLE IV
IMPACT OF MODEL STRUCTURAL COMPONENTS

Model Variant (SNR = 10 dB)	**Params (k)**	**NMSE (dB)**
With BatchNorm	113.6	−8.19
Kernel 5×5	310.8	−8.39
Narrow Width (32ch)	28.4	−7.74
Proposed (64ch, 3×3, w/o BN)	**113.1**	**−8.29**

at $p = 9$, and then slightly increases as p grows toward W. For M, the NMSE decreases from $M = 2$ to a minimum at $M = 10$. These trends confirm the existence of optimal p and M, proving **Remark 1** and **Remark 2**.

2) Impact of Inner-Loop Iterations: As illustrated in Fig. 15, the NMSE performance under the BGIN scenario exhibits a "U-shaped" trajectory. The method achieves rapid convergence, reaching the minimum error at a small number of steps. Notably, further increasing K leads to performance degradation, which can be attributed to the model overfitting the impulsive outliers in the noisy samples. This observation empirically validates our hypothesis in **Remark 3** that a small K is preferable for robust adaptation.

3) Impact of the Balancing Weight within the Composite Loss: Fig. 16 presents the sensitivity analysis under Gaussian and BGIN scenarios by varying the weight assigned to the $\mathcal{L}_{CR}$ from 0.1 to 0.9 at SNR $= 5$ dB. Results indicate that performance degrades at extreme weights in both scenarios. Notably, low weights cause severe deterioration under both noise, confirming the necessity of the $\mathcal{L}_{CR}$. Consequently, a weight of 0.5 is adopted to ensure robust generalization across diverse environments.

4) Impact of Network Depth: Fig. 17 illustrates the trade-off between network depth and performance at SNR $= 10$ dB. Shallow networks ($2-3$ layers) underperform due to limited capacity, while extending beyond 4 layers yields negligible gains despite linear parameter growth. Thus, the proposed 4-layer architecture strikes the optimal balance between accuracy and complexity.

5) Impact of Model Structural Components: Table IV summarizes the impact of various structural configurations on NMSE performance. First, incorporating BN degrades the NMSE from -8.29 dB to -8.19 dB, as normalization layers tend to restrict the range flexibility required for low-level signal recovery. Second, while increasing the kernel size to 5×5 yields a marginal gain, it nearly triples the parameter count, confirming that 3×3 kernels offer the optimal efficiency trade-off. Third, reducing the feature width to 32 channels leads to a significant performance drop, justifying the selection of 64 channels for sufficient representational capacity. Finally, residual connections are omitted since the shallow network depth avoids gradient vanishing, making additional skip connections computationally redundant.

TABLE V
IMPACT OF ALGORITHM COMPONENTS

Method / Ablation (SNR = 10 dB)	**NMSE (dB)**
(i) Diagonal Downsampling [41], [42]	-6.49
(ii) Bernoulli Masking [26], [36]	-6.12
(iii) w/o Sampling-Consistency Term $\mathcal{L}_{SC}$	-7.68
Proposed	**-8.29**

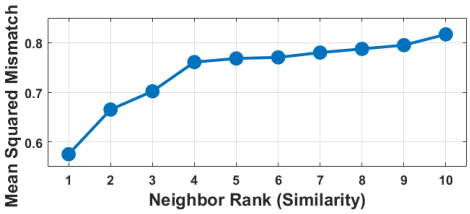

Fig. 18. Quantification of patch mismatch with respect to neighbor rank.

6) Impact of Algorithm Components: Table V isolates the contributions of specific design choices. Replacing the non-local pixel bank with diagonal downsampling or Bernoulli masking degrades NMSE to -6.49 dB and -6.12 dB, respectively. This confirms that structure-agnostic masking destroys critical pilot information, whereas the proposed non-local search effectively aggregates correlated features. Furthermore, removing the $\mathcal{L}_{SC}$ reduces performance to -7.68 dB, validating its role in ensuring training stability and robust generalization alongside the $\mathcal{L}_{CR}$.

7) Analysis of Patch Matching Mismatch: We further investigated the potential mismatch error introduced by the non-local pixel bank construction. Fig. 18 quantifies the deviation of matched patch centers from the ground truth across neighbor ranks. While the mismatch naturally increases from Rank 1 to 10, it stabilizes within the selected range ($M = 10$) without divergence. This confirms that the neighbors maintain structural relevance, and aggregating them effectively reduces estimation variance in low-SNR regimes. Crucially, the proposed $\mathcal{L}_{sc}$ acts as a regularization term that mitigates these local matching mismatches by enforcing global consistency.

8) Impact of Optimization Strategy and Sensitivity Analysis: Table VI compares the proposed Meta-SGD against standard SGD and AdamW (with cosine schedule), revealing a significant performance advantage. This confirms that the learned, per-parameter step size vector $\boldsymbol{\eta}$ provides superior adaptation capabilities over scalar-adaptive schemes in the sparse pilot landscape. Furthermore, replacing $\boldsymbol{\eta}$ with fixed scalars demonstrates high sensitivity. These results validate the necessity of the Meta-SGD mechanism for ensuring robust convergence without manual hyperparameter tuning.

VI. DISCUSSION AND FUTURE WORK

While this work focuses on block-fading channels, extending the ZS-SS framework to fast-fading scenarios represents a promising direction. The online adaptation capability of Meta-SGD could potentially be leveraged to track intra-block variations by incorporating temporal correlations, which we plan to explore in future research.

TABLE VI
COMPARISON OF OPTIMIZERS AND SENSITIVITY TO LEARNING RATE (η)

Optimization Strategy (SNR = 10 dB)	**NMSE (dB)**
SGD (Baseline)	-5.37
AdamW + Cosine Schedule	-5.71
Fixed $\eta = 0.0001$	-6.80
Fixed $\eta = 0.001$	-7.90
Fixed $\eta = 0.01$	-5.46
Proposed (Meta-SGD with Learned η)	**-8.29**

To address more complex multi-user environments, such as joint multi-user MIMO (MU-MIMO) estimation and extremely severe pilot contamination, future work will extend the proposed framework by incorporating additional physical priors and leveraging cross-cell cooperation mechanisms.

Future work also aims to bridge the latency gap by deploying our lightweight estimator as a "Teacher" in O-RAN Non-RT RICs [55]. By distilling knowledge into specially designed "Student" networks, we can better combine zero-shot accuracy with real-time inference capabilities.

VII. CONCLUSION

In this paper, we propose a lightweight ZS-SS learning framework with meta-learning for channel estimation in massive MIMO. It exploits non-local self-similarity by building a coefficient bank from LS pre-estimate and sampling coefficients to generate pseudo-training pairs. A compact CNN with a specially designed composite loss enables fast convergence, strong robustness and low complexity. To improve adaptability, we adopt Meta-SGD for online adaptation under dynamic channels. Simulations show significant performance gains over traditional and deep baselines in both Gaussian and non-Gaussian noise environments. Additionally, our method exhibited outstanding computational efficiency and real-time feasibility, highlighting its suitability for practical resource-constrained massive MIMO deployments.

APPENDIX A
THEORETICAL ANALYSIS OF LOSS FUNCTION DESIGN

For Gaussian and non-Gaussian noise with zero mean, consider two independent noisy observations, $Y^{(k_1)} = h + \varepsilon^{(k_1)}$ and $Y^{(k_2)} = h + \varepsilon^{(k_2)}$, where $\mathbb{E}[\varepsilon^{(k_1)}\varepsilon^{(k_2)*}] = 0$ for $k_1 \neq k_2$. Since both terms in (20) have the same expectation, we only expand the first:

$$\begin{aligned}\mathbb{E}\left[\left\|f(Y^{(k_1)}) - Y^{(k_2)}\right\|^2\right] &= \mathbb{E}\left[\left\|f(Y^{(k_1)}) - h\right\|^2\right] + \\ &\quad \mathbb{E}\left[\left\|\varepsilon^{(k_2)}\right\|^2\right] - \\ &\quad 2\,\mathrm{Re}\,\mathbb{E}\left[\left\langle f(Y^{(k_1)}) - h,\ \varepsilon^{(k_2)}\right\rangle\right] \\ &= \mathbb{E}\left[\left\|f(Y^{(k_1)}) - h\right\|^2\right] + \\ &\quad \mathbb{E}\left[\left\|\varepsilon^{(k_2)}\right\|^2\right],\end{aligned} \tag{34}$$

where $\mathbb{E}[\varepsilon^{(k_i)} \mid h] = 0$, the added term $\mathbb{E}\left[\|\varepsilon^{(k_2)}\|^2\right]$ is a constant independent of θ, hence minimising the $\mathcal{L}_{CR}$ is

equivalent to minimising the supervised MSE $\mathbb{E}\| f(Y)-h\|^2$. This ensures that the optimum is attained at the ground truth h. The averaging factor $\frac{1}{2}$ in (20) normalizes the loss and ensures the estimator is unbiased over both directions. Consequently, the gradient scale does not depend on the number of pairs, $\mathcal{L}_{CR}(\theta) = \frac{1}{2}\sum_{i=1}^{2}\ell_i$, where each ℓ_i is a loss term for one direction.

For more general non-Gaussian noise with non-zero mean $\mathbb{E}[\varepsilon^{(k_i)} \mid h] \neq 0$, each noisy observation can be decomposed as $Y^{(k_i)} = h + \mu + \tilde{\varepsilon}^{(k_i)}$, where $\mathbb{E}[\tilde{\varepsilon}^{(k_i)} \mid h] = 0$ and $\mu = \mathbb{E}[\varepsilon^{(k_i)} \mid h]$ is the bias. Under the $\mathcal{L}_{CR}$, the minimizer satisfies $f(Y) = h+\mu$, i.e., the estimator recovers the ground truth up to the mean bias μ. The resulting MSE becomes $\mathbb{E}\|f(Y)-h\|^2 = \|\mu\|^2$. When compared to the error of using the raw noisy observation Y, $\mathbb{E}\|Y-h\|^2 = \|\mu\|^2+\mathbb{E}\|\tilde{\varepsilon}\|^2$, we have $\mathbb{E}\|f(Y)-h\|^2 \leq \mathbb{E}\|Y-h\|^2$. That is, the $\mathcal{L}_{CR}$ always reduces the random error component, even under non-zero-mean perturbations. The variance from the random noise is always reduced, which explains the observed improvement in NMSE in practical scenarios.

Similarly, the $\mathcal{L}_{SC}$ in (23) can be equivalently written as

$$\mathcal{L}_{SC} = \frac{1}{2}\sum_{i=1}^{2}\left\|f(\mathbf{B}_{k_i}(Y)) - \mathbf{B}_{k_i}(f(Y))\right\|_2^2, \tag{35}$$

where $Y_i = \mathbf{B}_{k_i}(Y)$ and $\widehat{X}^{(k_i)} = \mathbf{B}_{k_i}(f(Y))$ as defined in (14) and (22). This term enforces

$$f(\mathbf{B}_{k_i}(Y)) \approx \mathbf{B}_{k_i}(f(Y)), \tag{36}$$

which means the network output is invariant under the order of denoising and patch sampling. Unlike the $\mathcal{L}_{CR}$, $\mathcal{L}_{SC}$ does not directly penalize the distance to the ground truth h, but only encourages consistency between different sampling routes. Therefore, its main role is to regularize the model and stabilize training, rather than to provide a direct denoising signal. Here the $\mathcal{L}_{SC}$ is nearly insensitive to μ since the bias μ appears identically in both branches, and primarily penalizes discrepancies caused by the random noise $\tilde{\varepsilon}$. This insensitivity is beneficial since it ensures the regularization effect of $\mathcal{L}_{SC}$ remains stable even under non-zero-mean perturbations, preventing bias amplification and focusing on suppressing structural drift between samples.

APPENDIX B
BIAS QUANTIFICATION AND CONSISTENCY ANALYSIS

While variance reduction is achieved via averaging, estimation accuracy depends on the bias from neighbor aggregation, particularly under non-zero-mean noise conditions (i.e., $\mathbb{E}[\varepsilon] \neq 0$). Let h be the target ground truth and $\mathcal{S}_M = \{h_i\}_{i=1}^{M}$ be the set of M nearest neighbors identified by the network. The estimator is modeled as $\hat{h} = \frac{1}{M}\sum_{i=1}^{M} Y_i$.

To quantify bias explicitly, we decompose the observation Y_i into signal mismatch $\Delta_i = h_i - h$ and noise components:

$$\varepsilon_i = \mu + \tilde{\varepsilon}_i, \tag{37}$$

where $\mu = \mathbb{E}[\varepsilon_i] \neq 0$ represents the systematic bias and $\tilde{\varepsilon}_i$ is the zero-mean random fluctuation ($\mathbb{E}[\tilde{\varepsilon}_i] = 0$). Taking the expectation $\mathbb{E}[\hat{h}]$ eliminates the random noise, yielding the bias decomposition:

$$\mathbb{E}[\hat{h}] - h = \underbrace{\frac{1}{M}\sum_{i=1}^{M}\Delta_i}_{\text{Patch Mismatch}} + \underbrace{\mu}_{\text{Systematic Bias}}. \tag{38}$$

This derivation confirms that when $\mathbb{E}[\varepsilon] \neq 0$, the systematic bias μ persists and adds to the search error. Let $\delta_{\max} = \max_i |h_i - h|$ be the maximum structural dissimilarity allowed by the search. Applying the triangle inequality to (38) yields the explicit bound:

$$\left|\mathbb{E}[\hat{h}] - h\right| \leq \frac{1}{M}\sum_{i=1}^{M}|\Delta_i| + |\mu| \leq \delta_{\max} + |\mu|. \tag{39}$$

The total bias is thus bounded by the worst-case mismatch $\delta_{\max}$ and the magnitude of the noise bias $|\mu|$. In our framework, minimizing $\mathcal{L}_{SC}$ tightens this bound by reducing $\delta_{\max}$.

Following non-local means theory [56], the estimator achieves consistency (i.e., $\hat{h} \xrightarrow{p} h$ and $\mathrm{Var}(\hat{h}) \to 0$) under specific asymptotic conditions. Specifically, as the bank size $N \to \infty$, the probability of finding exact matches approaches unity, implying that the structural mismatch $\delta_{\max} \to 0$. Simultaneously, the number of aggregated neighbors M must grow such that $M \to \infty$ and $M/N \to 0$, ensuring the variance of the random noise component scales as $\frac{1}{M}\mathrm{Var}(\tilde{\varepsilon}) \to 0$. Furthermore, for the case where $\mathbb{E}[\varepsilon] \neq 0$, consistency necessitates explicit bias removal such that the effective systematic bias $|\mu| \to 0$. Under these combined conditions, both the patch mismatch and noise components vanish, recovering the true channel h.

REFERENCES

[1] Z. Gao, W. Yi, F. Benkhelifa, and A. Nallanathan, "Meta-ZSN2N: Zero-shot learning for channel estimation in massive MIMO systems," in *2025 IEEE Global Commun. Conf. (GLOBECOM)*. IEEE, 2025, pp. 1–1.

[2] L. Lu, G. Y. Li, A. L. Swindlehurst, A. Ashikhmin, and R. Zhang, "An overview of massive MIMO: Benefits and challenges," *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 5, pp. 742–758, 2014.

[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, 2014.

[4] M. Jian, G. C. Alexandropoulos, E. Basar, C. Huang, R. Liu, Y. Liu, and C. Yuen, "Reconfigurable intelligent surfaces for wireless communications: Overview of hardware designs, channel models, and estimation techniques," *Intell. Converged Networks*, vol. 3, no. 1, pp. 1–32, 2022.

[5] J. Zou, W. Xie, J. Xiao, Y. Liang, J. M. Moualeu, and L. Yang, "From analog to digital semantic communications: Architectures, challenges, and future directions," *IEEE Wireless Commun.*, vol. 32, no. 5, pp. 56–62, 2025.

[6] C. Luo, J. Ji, Q. Wang, X. Chen, and P. Li, "Channel state information prediction for 5G wireless communications: A deep learning approach," *IEEE Trans. Network Sci. Eng.*, vol. 7, no. 1, pp. 227–236, 2018.

[7] L. Sun, T. Xu, S. Yan, J. Hu, X. Yu, and F. Shu, "On resource allocation in covert wireless communication with channel estimation," *IEEE Trans. on Commun.*, vol. 68, no. 10, pp. 6456–6469, 2020.

[8] A. Sufyan, K. B. Khan, O. A. Khashan, T. Mir, and U. Mir, "From 5G to beyond 5G: A comprehensive survey of wireless network evolution, challenges, and promising technologies," *Electronics*, vol. 12, no. 10, p. 2200, 2023.

[9] H. Ma, J. Du, H. Wu, L. Xing, and R. Zheng, "A review of channel estimation research methods for massive MIMO systems," *Phys. Commun.*, p. 102632, 2025.

[10] A. Melgar, A. de la Fuente, L. Carro-Calvo, Ó. Barquero-Pérez, and E. Morgado, "Deep neural network: an alternative to traditional channel estimators in massive MIMO systems," *IEEE Trans. Cognit. Commun. Netw.*, vol. 8, no. 2, pp. 657–671, 2022.

[11] J. Zou, J. Xiao, Q. Mao, S. Liu, B. Xiao, and Y. Liang, "Deep receiver for multi-layer data transmission with superimposed pilots," in *IEEE Int. Conf. Acoust. Speech Signal Process (ICASSP)*. IEEE, 2025, pp. 1–5.

[12] H. Kim, J. Choi, and D. J. Love, "Massive MIMO channel prediction via meta-learning and deep denoising: Is a small dataset enough?" *IEEE Trans. Wireless Commun.*, vol. 22, no. 12, pp. 9278–9290, 2023.

[13] H. Q. Ngo, E. G. Larsson, and T. L. Marzetta, "Energy and spectral efficiency of very large multiuser MIMO systems," *IEEE Trans. on Commun.*, vol. 61, no. 4, pp. 1436–1449, 2013.

[14] E. Balevi, A. Doshi, and J. G. Andrews, "Massive MIMO channel estimation with an untrained deep neural network," *IEEE Trans. Wireless Commun.*, vol. 19, no. 3, pp. 2079–2090, 2020.

[15] J. Wang, X. Mei, J. Xiao, X. Li, P. Zhu, A. Nallanathan, and C. Yuen, "Deep learning based wavenumber domain channel estimation for holographic MIMO communications," *IEEE Trans. Veh. Technol.*, no. 99, pp. 1–6, 2025.

[16] J. Xiao, J. Wang, Z. Wang, J. Wang, W. Xie, and Y. Liu, "Multi-task learning for near/far field channel estimation in STAR-RIS networks," *IEEE Trans. on Commun.*, vol. 72, no. 10, pp. 6344–6359, 2024.

[17] Z. Chen, H. Shin, and A. Nallanathan, "Generative diffusion model-based variational inference for MIMO channel estimation," *IEEE Trans. on Commun.*, 2025.

[18] H. Ye, G. Y. Li, and B.-H. Juang, "Deep learning based end-to-end wireless communication systems without pilots," *IEEE Trans. Cogn. Commun. Netw.*, vol. 7, no. 3, pp. 702–714, 2021.

[19] J. Xiao, J. Wang, Z. Wang, W. Xie, and Y. Liu, "Multi-scale attention based channel estimation for RIS-aided massive MIMO systems," *IEEE Trans. Wireless Commun.*, vol. 23, no. 6, pp. 5969–5984, 2023.

[20] S. Liu, Z. Gao, J. Zhang, M. Di Renzo, and M.-S. Alouini, "Deep denoising neural network assisted compressive channel estimation for mmWave intelligent reflecting surfaces," *IEEE Trans. Veh. Technol.*, vol. 69, no. 8, pp. 9223–9228, 2020.

[21] W. Chen, R. Yu, and Z. Ye, "An attention-based denoising neural network for mmWave RIS-assisted SIMO channel estimation," *IEEE Wireless Commun. Lett.*, vol. 13, no. 7, pp. 1933–1937, 2024.

[22] J. Guo, T. Guo, M. Li, T. Wu, and H. Lin, "Underwater-acoustic-OFDM channel estimation based on deep learning and data augmentation," *Electronics*, vol. 13, no. 4, p. 689, 2024.

[23] Q. Zhao, X. Zeng, Z. Fan, Q. Zhang, and W. Li, "Channel estimation for FDD massive MIMO with complex residual denoising network," *IEEE Wireless Commun. Lett.*, vol. 13, no. 8, pp. 2070–2074, 2024.

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

[25] Y. Li, X. Bian, and M. Li, "Denoising generalization performance of channel estimation in multipath time-varying OFDM systems," *Sensors*, vol. 23, no. 6, p. 3102, 2023.

[26] K. Kang, Q. Hu, Y. Cai, and Y. C. Eldar, "One-shot learning for channel estimation in massive MIMO systems," in *2023 IEEE 97th Veh. Technol. Conf.* IEEE, 2023, pp. 1–5.

[27] P. Chaudhary, I. Chauhan, B. Manoj, and M. R. Bhatnagar, "Linear regression-based channel estimation for non-Gaussian noise," in *2024 IEEE 99th Veh. Technol. Conf. (VTC2024-Spring)*. IEEE, 2024, pp. 1–6.

[28] G. Gui, L. Xu, and N. Shimoi, "Stable sparse channel estimation algorithm under non-Gaussian noise environments," in *2015 21st Asia-Pacific Conf. Commun. (APCC)*. IEEE, 2015, pp. 561–565.

[29] D. Middleton, "Non-Gaussian noise models in signal processing for telecommunications: new methods an results for class A and class B noise models," *IEEE Trans. Inf. Theory*, vol. 45, no. 4, pp. 1129–1149, 2002.

[30] H. Shao and N. C. Beaulieu, "An investigation of block coding for Laplacian noise," *IEEE Trans. Wireless Commun.*, vol. 11, no. 7, pp. 2362–2372, 2012.

[31] H. Meng, Y. L. Guan, and S. Chen, "Modeling and analysis of noise effects on broadband power-line communications," *IEEE Trans. on Power Delivery*, vol. 20, no. 2, pp. 630–637, 2005.

[32] K. J. Kerpez and A. M. Gottlieb, "The error performance of digital subscriber lines in the presence of impulse noise," *IEEE Trans. on Commun.*, vol. 43, no. 5, pp. 1902–1905, 2002.

[33] C. Zhao, J. Wang, W. Huang, X. Chen, and T. Qi, "Communication under mixed Gaussian-impulsive channel: an end-to-end framework," in *2024 IEEE Int. Conf. Mach. Learn. for Commun. and Netw. (ICMLCN)*. IEEE, 2024, pp. 119–125.

[34] C. Zhao, J. Wang, Q. Peng, W. Huang, X. Chen, Z. Zhao, and T. Le-Ngoc, "Deep learning based transceiver design for additive non-Gaussian impulsive noise channels," *IEEE Trans. Cogn. Commun. Netw.*, pp. 1–1, 2025.

[35] N. Sadeghi and M. Azghani, "Deep learning based channel estimation in PLC systems," *Ann. Telecommun.*, pp. 1–10, 2024.

[36] Y. Quan, M. Chen, T. Pang, and H. Ji, "Self2self with dropout: Learning self-supervised denoising from single image," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2020, pp. 1890–1898.

[37] S. P. Herath, N. H. Tran, and T. Le-Ngoc, "On optimal input distribution and capacity limit of Bernoulli-Gaussian impulsive noise channels," in *2012 IEEE Int. Conf. Commun. (ICC)*. IEEE, 2012, pp. 3429–3433.

[38] F. Yilmaz and M.-S. Alouini, "On the bit-error rate of binary phase shift keying over additive white generalized Laplacian noise (AWGLN) channels," in *2018 26th Signal Process. Commun. Appl. Conf. (SIU)*. IEEE, 2018, pp. 1–4.

[39] S. Niranjayan and N. C. Beaulieu, "Analysis of wireless communication systems in the presence of non-Gaussian impulsive noise and Gaussian noise," in *2010 IEEE Wireless Commun. Netw. Conf. (WCNC)*. IEEE, 2010, pp. 1–6.

[40] B. Clerckx and C. Oestges, *MIMO wireless networks: channels, techniques and standards for multi-antenna, multi-user and multi-cell systems*. Academic Press, 2013.

[41] Y. Mansour and R. Heckel, "Zero-Shot Noise2Noise: Efficient image denoising without any data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2023, pp. 14 018–14 027.

[42] J. Bai, D. Zhu, and M. Chen, "Dual-sampling Noise2Noise: efficient single-image denoising," *IEEE Trans. Instrum. Meas.*, vol. 74, pp. 1–12, 2025.

[43] S. K. Mohammed and S. Singh, "ZS-ACL: Light-weight zero-shot image denoising using alpha-conditional Loss," in *2024 39th Int. Conf. Image Vis. Comput. N. Z. (IVCNZ)*, 2024, pp. 1–6.

[44] Z. Gao, A. Junejo, W. Yi, A. Nallanathan, and F. Benkhelifa, "Zero-shot self-supervised channel estimation in massive MIMO LEO satellites systems," in *2025 IEEE Global Commun. Conf. (GLOBECOM) Workshops*. IEEE, 2025, pp. 1–1.

[45] Z. Zha, X. Yuan, J. Zhou, C. Zhu, and B. Wen, "Image restoration via simultaneous nonlocal self-similarity priors," *IEEE Trans. Image Process.*, vol. 29, pp. 8561–8576, 2020.

[46] W. Hu, Z. Fu, and Z. Guo, "Local frequency interpretation and non-local self-similarity on graph for point cloud inpainting," *IEEE Trans. Image Process.*, vol. 28, no. 8, pp. 4087–4100, 2019.

[47] H. Chen, X. He, L. Qing, and Q. Teng, "Single image super-resolution via adaptive transform-based nonlocal self-similarity modeling and learning-based gradient regularization," *IEEE Trans. Multimedia*, vol. 19, no. 8, pp. 1702–1717, 2017.

[48] Q. Ma, J. Jiang, X. Zhou, P. Liang, X. Liu, and J. Ma, "Pixel2Pixel: a pixelwise approach for zero-shot single image denoising," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 47, no. 6, pp. 4614–4629, 2025.

[49] J. Lehtinen, J. Munkberg, J. Hasselgren, S. Laine, T. Karras, M. Aittala, and T. Aila, "Noise2Noise: Learning image restoration without clean data," in *Int. Conf. Mach. Learn.*, 2018, pp. 4620–4631.

[50] Z. Gao, H. Liu, and L. Li, "Data augmentation for time-series classification: An extensive empirical study and comprehensive survey," *J. Artif. Intell. Res*, vol. 83, 2025.

[51] X. Chen and K. He, "Exploring simple siamese representation learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2021, pp. 15 750–15 758.

[52] Z. Li, F. Zhou, F. Chen, and H. Li, "Meta-SGD: Learning to learn quickly for few-shot learning," *arXiv preprint arXiv:1707.09835*, 2017.

[53] 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.

[54] I. J. Ratul, Y. Zhou, and K. Yang, "Accelerating deep learning inference: A comparative analysis of modern acceleration frameworks," *Electronics*, vol. 14, no. 15, p. 2977, 2025.

[55] B. Agarwal, R. Irmer, D. Lister, and G.-M. Muntean, "Open RAN for 6G networks: Architecture, use cases and open issues," *IEEE Commun. Surv. Tutorials*, 2025.

[56] A. Buades, B. Coll, and J.-M. Morel, "A review of image denoising algorithms, with a new one," *Multiscale Model. Simul.*, vol. 4, no. 2, pp. 490–530, 2005.