network

MDPI

*Article*

# A Coupled Multi-Stage Hybrid Framework for BER Prediction and Beam Angle Optimization in Massive MIMO Systems: Combining Classical Regression with Coupled Deep Learning Approaches

**Iacovos Ioannou [1,2,3,*], Michael Georgiades [4], Prabagarane Nagaradjane [5], Ala Khalifeh [6], Christophoros Christophorou [7], Marios Raspopoulos [7,8] and Vasos Vassiliou [2,3]**

1 Department of Computer Science, Philips University, 2001 Strovolos, Cyprus
2 Department of Computer Science, University of Cyprus, 1678 Nicosia, Cyprus; vasosv@ucy.ac.cy
3 CYENS Centre of Excellence, 1016 Nicosia, Cyprus
4 Department of Computer Science, Neapolis University Pafos, 8042 Pafos, Cyprus; m.georgiades@nup.ac.cy
5 Department of Electronics and Communication Engineering, SSN College of Engineering, Rajiv Gandhi Salai, Kalavakkam, Chennai 603110, India; prabagaranen@ssn.edu.in
6 School of Electrical Engineering and Information Technology, German Jordanian University, Amman 11180, Jordan; ala.khalifeh@gju.edu.jo
7 School of Sciences, UCLan Cyprus, 12-14 University Avenue, 7080 Pyla, Cyprus; cchristoforou2@uclan.ac.uk (C.C.); mraspopoulos@uclan.ac.uk (M.R.)
8 INSPIRE Research Centre, UCLan Cyprus, 12-14 University Avenue, 7080 Pyla, Cyprus
* Correspondence: ioannou.iakovos@philipsuni.ac.cy

## Abstract

A coupled multi-stage learning framework is presented for joint bit error rate (BER) prediction and beam angle optimization in massive multiple-input multiple-output (MIMO) systems under a controlled simulation protocol. Unlike purely sequential benchmarking pipelines, the proposed method jointly coordinates BER prediction and beam-angle selection through a shared latent representation, an uncertainty-guided refinement mechanism, a cross-stage consistency loss and alternating optimization. Ten diverse approaches are systematically evaluated across two task-specific stages: Stage 1 examines six classical and adapted methods for BER prediction, including polynomial regression and deep unfolding networks; Stage 2 investigates four machine-learning and generative adversarial network (GAN)-based approaches for angle optimization, including conditional GANs and the proposed Direct-Angle neural network. Stage 3 couples the best-performing methods into a unified hybrid architecture through a shared encoder, explicit consistency regularization and alternating cross-stage updates, thereby producing an integrated beamforming decision strategy rather than an independent cascade. It is shown through the evaluation that the coupled hybrid framework achieves 96.0% overall angle-selection accuracy, a mean BER of $8.0 \times 10^{-5}$ and 100% BER tolerance compliance within $\pm 3$ dB. In this framework, a differentiable BER surrogate initialized from a second-degree polynomial-regression teacher is coupled with the proposed Direct-Angle-NN for angle optimization. Relative to the strongest reimplemented literature baseline under the same controlled simulation assumptions, a 33.3% reduction in mean BER is achieved. Ablation experiments show that the coupling mechanism provides a modest but consistent improvement over the decoupled sequential baseline, increasing angle-selection accuracy from 93.5% to 96.0% and reducing mean BER from $1.05 \times 10^{-4}$ to $8.0 \times 10^{-5}$; the shared encoder accounts for the largest part of this gain while the consistency loss adds 0.6 percentage points. These results indicate that the shared encoder, consistency regularization and uncertainty-guided refinement improve the final beamforming decision, although the gain should be interpreted as incremental



Academic Editor: Alberto Gotta

Received: 22 March 2026
Revised: 18 May 2026
Accepted: 21 May 2026
Published: 27 May 2026

rather than as a large architectural breakthrough. A spectral efficiency of 38.0 bps/Hz and an energy efficiency of 0.466 Gbps/W are achieved with a power consumption of only 32.6 W. The theoretical discussion is presented as an analytical characterization of BER sensitivity, complemented by a computational-complexity assessment and empirical convergence diagnostics for the alternating optimization, rather than as a formal optimality proof. The effectiveness of the framework across multiple performance metrics is supported by Monte Carlo simulations, while the limitations of the current setup, including perfect CSI, uncoded QPSK, ideal hardware assumptions and a fixed beam codebook, are explicitly discussed. The complete simulation framework, including code and trained models, can be made available by the corresponding author upon reasonable request to facilitate reproducible research in massive MIMO optimization.

**Keywords:** massive MIMO; BER prediction; beam optimization; coupled learning framework; hybrid beamforming; deep learning; generative adversarial networks; 6G networks; machine learning; neural networks; spectral efficiency

## 1. Introduction

The evolution toward sixth-generation (6G) wireless networks demands unprecedented improvements in spectral efficiency, energy efficiency and communication reliability to support emerging applications, including extended reality, holographic communications and massive machine-type communications [1,2]. Massive multiple-input multiple-output (MIMO) systems represent a cornerstone technology for achieving these objectives by deploying large antenna arrays that enable sophisticated beamforming and spatial multiplexing capabilities [3,4]. The fundamental premise of massive MIMO is that by equipping base stations with hundreds or thousands of antenna elements, the system can simultaneously serve multiple users with high spectral efficiency while concentrating radiated energy toward intended receivers, thereby improving both throughput and energy efficiency.

However, optimal configuration of massive MIMO systems requires accurate prediction of bit error rate (BER) performance and precise beam angle optimization, tasks that become increasingly challenging as system complexity grows. The relationship between system configuration parameters (antenna weights, beam steering angles and power allocation) and the resulting BER performance is highly nonlinear, depending on complex interactions among channel conditions, interference patterns and hardware impairments [5]. Furthermore, the optimization landscape for beam angle selection is typically non-convex with multiple local optima, rendering gradient-based methods sensitive to initialization and prone to suboptimal solutions.

Traditional approaches to BER prediction and beam optimization rely on exhaustive search methods or analytical models that often fail to capture the complex, nonlinear relationships inherent in practical wireless channels [6]. Grid search baseline methods, while conceptually straightforward, suffer from prohibitive computational complexity that scales exponentially with the number of configurable parameters. For a system with $N$ parameters, each taking $M$ discrete values, exhaustive search requires $O(M^N)$ evaluations, rendering real-time optimization infeasible for practical massive MIMO configurations. Furthermore, analytical models based on simplified channel assumptions frequently exhibit significant prediction errors under realistic propagation conditions, where multipath fading, spatial correlation and hardware nonidealities introduce deviations from idealized behavior.

Recent advances in machine learning (ML) and deep learning (DL) have opened new avenues for addressing these challenges [7,8]. Generative adversarial networks (GANs)

have demonstrated remarkable capabilities in learning complex data distributions, making them attractive candidates for channel modeling and signal synthesis [9]. The adversarial training paradigm enables GANs to capture intricate statistical relationships that are difficult to model analytically, while the generative nature of these networks allows for efficient sampling of optimal configurations without exhaustive enumeration. Similarly, deep neural networks have shown promise in learning optimal beamforming strategies directly from data without requiring explicit channel models, effectively approximating the complex mapping from channel state information to optimal precoding matrices [10,11].

Despite these advances, existing approaches typically address BER prediction and beam optimization as separate problems, potentially missing synergies that could emerge from joint optimization. The BER achieved by a given beam configuration depends on how well the beam steering angles match the dominant propagation paths, while the optimal beam angles depend on the BER objectives that define the performance criteria. This interdependence suggests that a unified framework addressing both tasks simultaneously could achieve superior performance compared with sequential or independent optimization.

Unlike a purely sequential benchmarking pipeline, the proposed method introduces a ***coupled multi-stage learning framework*** in which BER prediction and angle selection are jointly coordinated through a shared latent representation, a cross-stage consistency constraint and alternating optimization. Specifically, the BER prediction module does not operate as an isolated preprocessing step. Instead, its output and uncertainty guide the angle-selection module while the resulting angle decisions are fed back to regularize BER estimation. Therefore, the contribution of this work is not a new physical MIMO hardware architecture, but a new ***learning-based coupled optimization framework*** for hybrid analog-digital beamforming configuration under the considered massive MIMO system model.

Furthermore, the relative performance of classical statistical methods versus modern deep learning approaches remains unclear, with limited systematic comparisons across diverse methodologies. While deep learning has achieved impressive results on many benchmark tasks, the added complexity and data requirements may not always be justified, particularly when classical methods can achieve comparable performance with greater interpretability and computational efficiency. A comprehensive evaluation spanning both paradigms is necessary to identify the most effective approaches for each subtask and to guide practical system design.

These gaps are addressed in this paper through a comprehensive coupled multi-stage hybrid framework in which the best approaches for joint BER prediction and beam angle optimization are systematically evaluated and combined. The framework is designed around three key observations that motivate its architecture:

1. Classical regression methods excel at capturing smooth, monotonic relationships with high interpretability and computational efficiency, whereas deep learning approaches can model complex nonlinear patterns and achieve superior generalization when sufficient training data is available. Through systematic evaluation of both paradigms, the most suitable approach for each subtask can be identified.
2. BER prediction and beam angle optimization exhibit different mathematical characteristics. BER prediction involves estimating a continuous output from channel features, a regression task well-suited to polynomial methods. Angle optimization involves finding discrete optimal configurations from a large search space, a combinatorial problem where learned heuristics can provide speedup over exhaustive search in large or continuous codebooks.
3. Combining the best methods for each subtask into a coupled framework enables cross-stage coordination while preserving the advantages of specialized approaches.

The hybrid architecture links both task-specific branches through a shared encoder, an explicit consistency loss, uncertainty-guided decision refinement and alternating optimization, moving beyond simple sequential chaining of independently optimized models.

Among these contributions, the Direct-Angle-NN architecture and the coupled multi-stage learning framework constitute the principal methodological advances of this paper, while OAMPNet-BER, ConformalBER and the Adaptive Bayesian Ensemble are adaptations of existing techniques to the BER-estimation context.

For clarity, the main contributions of this work are summarized in three points:

1. A coupled multi-stage hybrid framework is proposed for joint BER prediction and beam-angle selection in massive MIMO systems. The framework links the two subtasks through a shared latent representation, a cross-stage consistency loss, uncertainty-guided refinement and alternating optimization, thereby moving beyond a purely sequential pipeline.
2. A task-specific learning design is developed and evaluated across classical regression, ensemble learning, deep unfolding, conformal prediction, GAN-based learning and neural classification models. In particular, the Direct-Angle-NN architecture is introduced for beam-angle optimization using channel-aware attention and smoothness-regularized classification.
3. Ablation experiments quantify the incremental contribution of the shared encoder, consistency loss and uncertainty-guided refinement over a decoupled sequential baseline. The proposed framework is further validated through Monte Carlo simulations, literature comparison with paired bootstrap confidence intervals, an analytical discussion of BER sensitivity and a computational-complexity assessment that separates offline training cost from online inference cost. The coupling mechanism provides a modest but consistent improvement over the decoupled baseline rather than a large architectural breakthrough.

For reader convenience, a comprehensive list of mathematical symbols and notation used throughout this paper is provided in Appendix A and a complete glossary of abbreviations is given in Appendix B.

The remainder of this paper is organized as follows. Section 2 reviews related work and provides background on the methods employed in this study. Section 3 presents the system model and problem formulation. Section 4 details the proposed coupled multi-stage hybrid framework, including the coupled problem formulation, joint objective function, alternating optimization strategy and uncertainty-guided refinement. Section 5 presents comprehensive simulation results, ablation analysis and literature comparison. Finally, Section 6 concludes this paper and outlines directions for future research.

## 2. Literature Review and Background Work

A comprehensive review of the state of the art in machine learning-based massive MIMO optimization is provided in this section, together with background on the specific methods investigated in the proposed framework. Four key research areas are first examined: deep learning approaches for beam management, GAN-based methods for channel modeling and signal enhancement, physics-informed approaches and hybrid systems. Detailed background is then provided on all ten methods evaluated in the three-stage framework. Through this combined survey, the research gap addressed by this work is identified.

### *2.1. Deep Learning for Beam Management*

Deep learning approaches have gained significant traction for beam management in massive MIMO systems, driven by their ability to learn complex nonlinear mappings from high-dimensional inputs to optimal configurations. Liang et al. [12] presented a comprehensive survey of deep learning techniques for wireless resource allocation, including beam management and scheduling in vehicular networks. Their work demonstrated that deep neural networks can learn near-optimal resource allocation policies directly from channel observations, achieving significant speedups over iterative optimization solvers. By framing the beam management problem as a supervised or reinforcement learning task, their approach showed that learned policies generalize across varying network conditions, although training data requirements and computational overhead during the offline phase remain practical concerns.

Attiah et al. [13] proposed a deep learning framework for joint channel sensing and hybrid precoding in time-division duplex (TDD) massive MIMO OFDM systems. Their approach uses a deep neural network to map received uplink pilot signals directly to downlink hybrid precoders, bypassing explicit high-dimensional channel estimation entirely. This end-to-end design achieves competitive performance with substantially reduced pilot overhead compared with conventional two-step approaches that first estimate the channel and then compute the precoder. However, the approach assumes TDD reciprocity and does not address the frequency-division duplex (FDD) setting where downlink channel feedback is required.

Hu et al. [14] proposed a joint deep reinforcement learning and deep unfolding architecture for beam selection and precoding in millimeter-wave multiuser MIMO systems equipped with lens antenna arrays. Their method combines the adaptability of DRL for discrete beam selection with the efficiency of unfolded iterative algorithms for continuous precoding optimization, demonstrating improved sum-rate performance compared with purely DRL or purely model-based baselines. While the joint optimization yields significant gains, the convergence behavior of the DRL component in highly non-stationary environments and the computational cost of alternating between the two optimization modules remain practical challenges.

Recent studies have also examined interference-aware transmission and hardware-efficient beamforming in large-array systems. Ahmad and Shin [15] investigated massive MIMO NOMA with wavelet pulse shaping to reduce undesired channel interference, highlighting the importance of waveform-level interference mitigation in dense multiuser transmission. Nerini and Clerckx [16] studied analog computing for gigantic MIMO beamforming, showing that analog-domain processing can substantially reduce the computational burden of large-scale beamforming. These works reinforce the need for beamforming frameworks that consider both learning-based decision quality and practical implementation constraints, motivating the offline-vs-online cost separation reported in Section 5.6.4 of the present paper.

### *2.2. GAN-Based Channel Modeling and Enhancement*

Generative adversarial networks have emerged as powerful tools for wireless system design, leveraging the adversarial training paradigm to learn complex mappings and distributions from data. O'Shea and Hoydis [17] provided a foundational introduction to deep learning for the physical layer, demonstrating that autoencoder architectures, including GAN-inspired designs, can learn end-to-end communication system representations that jointly optimize modulation, coding and detection. Their work showed that neural network models can match or exceed the performance of hand-designed modulation schemes

under certain channel conditions, establishing deep learning as a viable paradigm for physical-layer design.

Balevi and Andrews [18] developed a deep learning-based channel estimation method for high-dimensional signals, proposing a denoising neural network that exploits the low-dimensional structure of wireless channels to achieve improved estimation accuracy under low signal-to-noise ratio (SNR) conditions. Their approach outperforms conventional linear minimum mean-squared error (LMMSE) estimators, particularly in scenarios where the channel exhibits complex spatial correlations that are difficult to capture with parametric models. However, the method requires retraining when the channel statistics change significantly.

Ye et al. [19] investigated end-to-end deep learning for wireless communication systems using conditional GANs to model unknown channels. By treating the physical channel as a conditional generative process, their framework enables joint transmitter and receiver optimization without requiring an explicit channel model. This approach is particularly relevant for scenarios where accurate analytical channel models are unavailable and the adversarial training captures complex channel behaviors, including nonlinear distortions and frequency-selective fading. However, GAN training instability and the need for careful hyperparameter balancing remain practical limitations.

### 2.3. Physics-Informed and Hybrid Approaches

Physics-informed neural networks (PINNs) have emerged as a promising approach for incorporating domain knowledge into learning frameworks, potentially improving generalization and reducing data requirements [20]. Kamal et al. [21] proposed HGGO-XCovNet, a hybrid optimization framework combining Hippo Graylag Goose Optimization (HGGO) with an Xception convolutional neural network (XCovNet) for energy-efficient resource allocation in MIMO-enabled wireless networks. Their approach jointly optimizes signal-to-interference-plus-noise ratio (SINR), data rate and power consumption, achieving improved energy efficiency compared with conventional allocation strategies. The integration of a metaheuristic optimizer with a deep learning model demonstrates the benefits of combining search-based and learning-based techniques, although the approach focuses on resource allocation rather than direct beamforming optimization.

Sohrabi et al. [22] proposed a deep learning framework for distributed channel feedback and multiuser precoding in FDD massive MIMO systems, where user terminals independently compress and feed back channel state information through learned encoders and the base station decodes the feedback to design multiuser precoders through a learned decoder network. By jointly training the user-side encoders and the base-station decoder in an end-to-end fashion, the approach significantly outperforms separate channel estimation and precoding designs, particularly under limited feedback bandwidth constraints. This work demonstrates the value of learning-based approaches for jointly optimizing distributed components that are traditionally designed independently.

Despite these advances, existing approaches predominantly address BER prediction and beam optimization as separate tasks, failing to exploit potential synergies from joint optimization. Furthermore, the relative merits of classical statistical methods versus deep learning approaches remain unclear owing to the absence of systematic comparisons across diverse methodologies. These gaps are addressed in this work through a comprehensive evaluation and a coupled hybrid framework design in which both subtasks are explicitly linked through shared representations and cross-stage consistency.

### 2.4. Background Work on Investigated Methods

The theoretical foundations, mathematical formulations and background for all ten methods evaluated in the proposed coupled multi-stage hybrid framework are provided in

this subsection. The six methods used for BER prediction in Stage 1 and the four methods used for angle optimization in Stage 2 are covered. The methodological rationale for selecting and combining these methods within the framework is presented in Section 4.

#### 2.4.1. Stage 1 Methods: BER Prediction

Linear regression is the most fundamental supervised learning method for continuous-valued prediction, establishing a direct affine mapping from input features to the target variable [23]. In the context of BER prediction, the model assumes that the system-level BER can be approximated as a linear combination of the extracted channel-feature vector $\boldsymbol{f}$:

$$\widehat{\mathrm{BER}} = \boldsymbol{\beta}^T \boldsymbol{f} + b, \tag{1}$$

where $\boldsymbol{\beta} \in \mathbb{R}^d$ is the learned regression weight vector and $b \in \mathbb{R}$ is the bias term. The model parameters are obtained by minimizing the residual sum of squares, yielding a closed-form solution via the normal equations. Despite its simplicity, linear regression provides an important lower-bound reference: any more complex method must demonstrably outperform this baseline to justify additional computational cost. Linear regression has been applied extensively in wireless communications for link-budget estimation and path-loss modeling [24] and its interpretability makes it valuable for validating the relevance of extracted features.

Polynomial regression extends linear regression by incorporating higher-order feature interactions, enabling the model to capture nonlinear relationships between channel features and BER without abandoning the interpretable, closed-form structure of linear models [23,25]. A second-degree polynomial augments the original feature vector with all pairwise products and squared terms:

$$\widehat{\mathrm{BER}} = \sum_i \beta_i f_i + \sum_{i \le j} \beta_{ij} f_i f_j + b, \tag{2}$$

where $\beta_i$ are first-order coefficients, $\beta_{ij}$ are second-order interaction coefficients and $b$ is the bias. The quadratic terms are particularly well-motivated for BER prediction because the BER-SINR relationship mediated by the Gaussian $Q$-function exhibits a smooth, monotonically decreasing curve whose local behavior is well approximated by low-order polynomials in the log domain [26].

Ridge regression augments ordinary least squares with an $\ell_2$ penalty on the regression coefficients, shrinking parameter magnitudes toward zero to reduce variance at the cost of a small increase in bias [23,27]. The regularized objective is

$$\boldsymbol{\beta}^* = \arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{N} \left(\mathrm{BER}_i - \boldsymbol{\beta}^T \boldsymbol{f}_i\right)^2 + \lambda_r \|\boldsymbol{\beta}\|_2^2, \tag{3}$$

where $\lambda_r > 0$ is the regularization parameter. Ridge regression is particularly effective when the feature matrix exhibits multicollinearity, which is common in massive MIMO feature extraction because channel power gains and inter-user correlations are inherently correlated.

Orthogonal approximate message passing (OAMP) is an iterative signal recovery algorithm originally developed for compressed sensing and MIMO detection [28]. The deep unfolding paradigm [29,30] converts fixed iterations of OAMP into layers of a neural network, with hand-crafted denoisers replaced by learnable nonlinear functions:

$$\boldsymbol{z}^{(t)} = \boldsymbol{f}_{\mathrm{oamp}} - \mathbf{A}^T \boldsymbol{r}^{(t-1)}, \tag{4}$$

$$\hat{\boldsymbol{x}}^{(t)} = \eta_\psi\left(\boldsymbol{z}^{(t)}\right), \tag{5}$$

$$\boldsymbol{r}^{(t)} = \boldsymbol{y}_{\mathrm{oamp}} - \mathbf{A}\hat{\boldsymbol{x}}^{(t)}, \tag{6}$$

where $\mathbf{A}$ is the measurement matrix, $\boldsymbol{r}^{(t)}$ is the residual at iteration $t$ and $\eta_\psi(\cdot)$ is a learnable denoising function parameterized by neural network weights $\psi$. The OAMPNet-BER variant adapts this concept to BER estimation, in which the unfolded iterations progressively refine a BER estimate rather than recovering a transmitted signal vector.

Conformal prediction is a distribution-free framework for constructing prediction intervals with guaranteed finite-sample coverage [31,32]. Given a base predictor and a calibration dataset, conformal methods produce intervals satisfying

$$P\Big(\mathrm{BER} \in [\widehat{\mathrm{BER}}_L, \widehat{\mathrm{BER}}_U]\Big) \geq 1 - \alpha, \tag{7}$$

where $\alpha \in (0, 1)$ is the user-specified significance level. In wireless communications, calibrated uncertainty estimates are valuable for resource allocation and link adaptation.

Bayesian model averaging (BMA) provides a principled framework for combining predictions from multiple models by weighting each according to its posterior probability [33,34]. The ensemble prediction is

$$\widehat{\mathrm{BER}} = \sum_{m=1}^{M_e} \omega_m \cdot \widehat{\mathrm{BER}}_m, \tag{8}$$

where $\omega_m \propto P(\mathcal{D}|M_m)$ is the Bayesian model weight proportional to the marginal likelihood of model $M_m$.

#### 2.4.2. Stage 2 Methods: Angle Optimization

Random forests aggregate predictions of multiple decision trees, each trained on a bootstrap sample with random feature subsets [35]. In the implementation used here, the Random Forest predicts a normalized beam-angle value, which is then projected onto the nearest entry of the finite codebook $\mathcal{C}$:

$$\hat{\theta}^* = \Pi_{\mathcal{C}}\left(\frac{1}{T}\sum_{t=1}^{T}\hat{\theta}_t^*\right), \tag{9}$$

where $T$ is the number of trees, $\hat{\theta}_t^*$ is the prediction of tree $t$ and $\Pi_{\mathcal{C}}(\cdot)$ denotes nearest-codebook projection.

The MLP is a feedforward neural network with one or more hidden layers and nonlinear activations [36,37]. For the baseline angle-optimization experiment, the MLP predicts a normalized beam-angle value and the output is snapped to the nearest codebook entry, matching the regression-to-codebook protocol used by the Random Forest baseline. Dropout regularization [38] is applied to prevent overfitting.

Conditional GANs extend the original GAN framework [9] by conditioning both the generator and discriminator on auxiliary information [39]. The following equation presents the generic conditional GAN background objective; the actual cGAN-BER implementation used in Stage 2 produces BER estimates with a separate auxiliary MLP for angle prediction, as described in Section 4.2.

$$\min_G \max_D \ \mathbb{E}\big[\log D(\theta^* \mid \boldsymbol{f})\big] + \mathbb{E}\big[\log\big(1 - D(G(\boldsymbol{\xi} \mid \boldsymbol{f}) \mid \boldsymbol{f})\big)\big], \tag{10}$$

where $G$ is the generator that produces candidate angle configurations from noise $\boldsymbol{\xi}$ conditioned on channel features $\boldsymbol{f}$ and $D$ is the discriminator that distinguishes real optimal angles from generated ones.

Attention mechanisms enable neural networks to dynamically weight input features based on learned relevance scores [40,41]. The proposed Direct-Angle-NN employs a channel-aware attention mechanism:

$$f' = \text{softmax}(\mathbf{W}_a f) \odot f, \tag{11}$$

where $\mathbf{W}_a \in \mathbb{R}^{d \times d}$ is the learnable attention weight matrix and $\odot$ denotes the element-wise product. The network is trained with a composite loss:

$$\mathcal{L}_{\text{DA}} = -\sum_{i=1}^{N_{\text{ang}}} y_i \log(\hat{y}_i) + \lambda_{\text{smooth}} \sum_{i=1}^{N_{\text{ang}}-1} (\hat{y}_{i+1} - \hat{y}_i)^2, \tag{12}$$

where the second term is a finite-difference smoothness penalty over adjacent codebook-class logits, through which the physical structure of the beam codebook is exploited, since neighbouring codewords steer to nearby directions.

## 3. System Model

A detailed description of the considered massive MIMO system architecture and the corresponding mathematical models is provided in this section. The considered downlink hybrid analog-digital beamforming configuration is illustrated in Figure 1. The physical transceiver architecture follows a standard hybrid massive MIMO model widely used in the literature; therefore, the novelty of this work lies not in proposing a new hardware architecture, but in developing a coupled learning framework that jointly coordinates BER prediction and beam-angle selection under this system model.

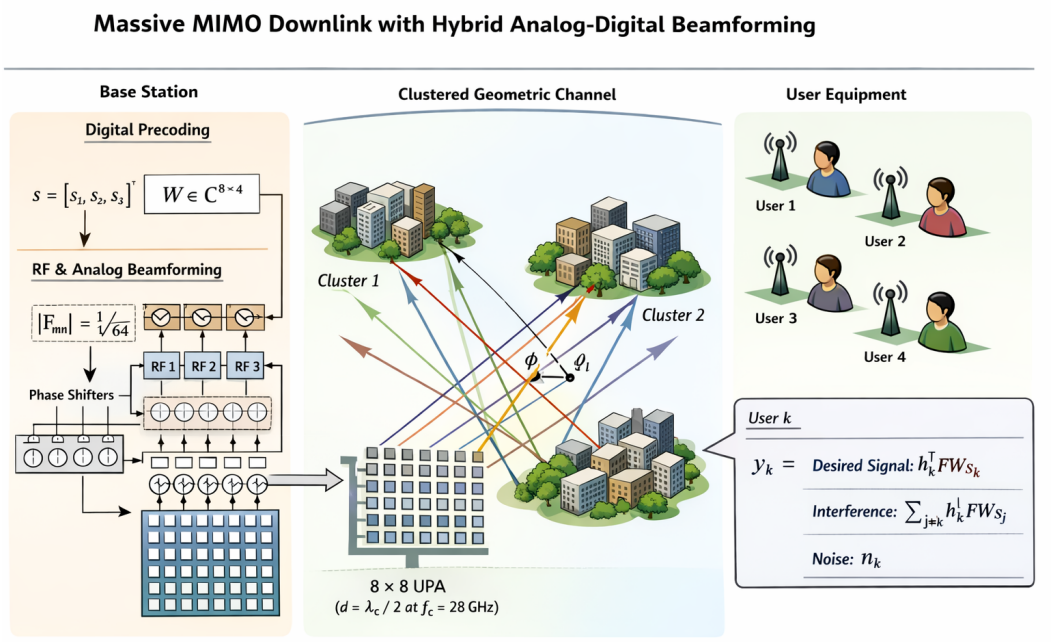


**Figure 1.** System model: massive MIMO downlink with hybrid analog-digital beamforming. The base station employs $N_t = 64$ antennas in an $8 \times 8$ UPA configuration with $N_{\text{RF}} = 8$ RF chains, serving $K = 4$ single-antenna users through a clustered geometric channel at $f_c = 28$ GHz. Arrows indicate the downlink signal flow from the baseband digital precoder through the RF chains and the analog beamforming network to the served users.

### *3.1. Massive MIMO Architecture*

A massive MIMO downlink system is considered, with $N_t = 64$ transmit antennas at the base station serving $K = 4$ single-antenna users through hybrid analog-digital

beamforming. The hybrid architecture employs $N_{\mathrm{RF}} = 8$ RF chains. The antenna array is configured as a uniform planar array (UPA) with $8 \times 8$ elements and half-wavelength inter-element spacing $d = \lambda_c/2$, where $\lambda_c = c/f_c$ is the carrier wavelength and $f_c = 28\,\mathrm{GHz}$ is the operating frequency.

The transmitted signal vector is expressed as

$$\boldsymbol{x} = \mathbf{F}\mathbf{W}\boldsymbol{s}, \tag{13}$$

where $\boldsymbol{x} \in \mathbb{C}^{N_t \times 1}$ is the transmitted signal vector, $\boldsymbol{s} \in \mathbb{C}^{K \times 1}$ is the symbol vector with $\mathbb{E}[\boldsymbol{s}\boldsymbol{s}^H] = \mathbf{I}_K$, $\mathbf{W} \in \mathbb{C}^{N_{\mathrm{RF}} \times K}$ is the digital baseband precoding matrix satisfying $\|\mathbf{W}\|_F^2 = K$ and $\mathbf{F} \in \mathbb{C}^{N_t \times N_{\mathrm{RF}}}$ is the analog beamforming matrix.

The analog beamformer is constrained to have constant-modulus entries:

$$|[\mathbf{F}]_{m,n}| = \frac{1}{\sqrt{N_t}}, \quad \forall m \in \{1, \ldots, N_t\},\ n \in \{1, \ldots, N_{\mathrm{RF}}\}. \tag{14}$$

### 3.2. Channel Model

The channel between the base station and user $k$ follows a clustered geometric model:

$$\boldsymbol{h}_k = \sqrt{\frac{N_t}{N_{\mathrm{cl}} N_{\mathrm{ray}}}} \sum_{i=1}^{N_{\mathrm{cl}}} \sum_{l=1}^{N_{\mathrm{ray}}} \alpha_{i,l}^{(k)}\, \boldsymbol{a}\left(\phi_{i,l}^{(k)}, \theta_{i,l}^{(k)}\right), \tag{15}$$

where $N_{\mathrm{cl}} = 5$ is the number of scattering clusters, $N_{\mathrm{ray}} = 10$ is the number of rays per cluster, $\alpha_{i,l}^{(k)} \sim \mathcal{CN}(0,1)$ is the complex path gain and $\boldsymbol{a}(\phi, \theta)$ is the UPA array response vector:

$$\boldsymbol{a}(\phi, \theta) = \boldsymbol{a}_{\mathrm{az}}(\phi) \otimes \boldsymbol{a}_{\mathrm{el}}(\theta), \tag{16}$$

with components

$$[\boldsymbol{a}_{\mathrm{az}}(\phi)]_m = \frac{1}{\sqrt{N_{\mathrm{az}}}} e^{j\pi(m-1)\sin(\phi)}, \quad m = 1, \ldots, N_{\mathrm{az}}, \tag{17}$$

$$[\boldsymbol{a}_{\mathrm{el}}(\theta)]_n = \frac{1}{\sqrt{N_{\mathrm{el}}}} e^{j\pi(n-1)\sin(\theta)}, \quad n = 1, \ldots, N_{\mathrm{el}}, \tag{18}$$

where $N_{\mathrm{az}} = N_{\mathrm{el}} = 8$. The cluster angles are drawn from Laplacian distributions with angular spread $\sigma_\phi = \sigma_\theta = 7.5^\circ$.

### 3.3. Signal Model and SINR

The received signal at user $k$ is

$$y_k = \boldsymbol{h}_k^H \mathbf{F} \boldsymbol{w}_k s_k + \sum_{j \neq k} \boldsymbol{h}_k^H \mathbf{F} \boldsymbol{w}_j s_j + n_k, \tag{19}$$

where $n_k \sim \mathcal{CN}(0, \sigma_n^2)$ with $\sigma_n^2 = N_0 B$ and $B = 400\,\mathrm{MHz}$. The instantaneous SINR for user $k$ is

$$\gamma_k = \frac{|\boldsymbol{h}_k^H \mathbf{F} \boldsymbol{w}_k|^2}{\sum_{j \neq k} |\boldsymbol{h}_k^H \mathbf{F} \boldsymbol{w}_j|^2 + \sigma_n^2}. \tag{20}$$

### 3.4. BER Computation

For QPSK modulation with Gray coding:

$$\mathrm{BER}_k = Q\left(\sqrt{2\gamma_k}\right) \approx \frac{1}{2}\,\mathrm{erfc}\left(\sqrt{\gamma_k}\right). \tag{21}$$

The system-level BER is

$$\mathrm{BER}_{\mathrm{sys}} = \frac{1}{K}\sum_{k=1}^{K}\mathrm{BER}_k. \tag{22}$$

### *3.5. Optimization Problem Formulation*

$$\begin{aligned} \min_{\mathbf{F},\mathbf{W}} \quad & \mathrm{BER}_{\mathrm{sys}} \\ \text{s.t.} \quad & |[\mathbf{F}]_{m,n}| = \frac{1}{\sqrt{N_t}}, \quad \forall m,n, \\ & \|\mathbf{FW}\|_F^2 \le P_{\max}. \end{aligned} \tag{23}$$

### *3.6. Training Data Generation*

The feature vector for each sample comprises

$$\boldsymbol{f} = \left[\|\boldsymbol{h}_1\|^2, \ldots, \|\boldsymbol{h}_K\|^2,\ \mathrm{Re}(\boldsymbol{h}_k^H\boldsymbol{h}_j),\ \mathrm{Im}(\boldsymbol{h}_k^H\boldsymbol{h}_j), \ldots\right]^T. \tag{24}$$

Table 1 summarizes the system and simulation parameters. The complete simulated dataset contains 11,000 samples, divided into 8000 training samples, 1000 validation samples and 2000 test samples.

**Table 1.** System and simulation parameters.

| Parameter | Value |
|---|---|
| Number of transmit antennas ($N_t$) | 64 |
| Number of RF chains ($N_{\mathrm{RF}}$) | 8 |
| Number of users ($K$) | 4 |
| Carrier frequency ($f_c$) | 28 GHz |
| Signal bandwidth ($B$) | 400 MHz |
| Number of clusters ($N_{\mathrm{cl}}$) | 5 |
| Rays per cluster ($N_{\mathrm{ray}}$) | 10 |
| Angular spread ($\sigma_\phi$, $\sigma_\theta$) | 7.5° |
| SNR range | 0 to 30 dB |
| Modulation | QPSK |
| Beam codebook size ($N_{\mathrm{ang}}$) | 64 paired beams ($8 \times 8$ azimuth/elevation grid) |
| Angle grid spacing | 17.14° per axis over $[-60°, +60°]$ |
| Total simulated samples | 11,000 |
| Training samples | 8000 |
| Validation samples | 1000 |
| Test samples | 2000 |
| Training/validation/test split | 72.7%/9.1%/18.2% |
| Stage 1 methods evaluated | 6 |
| Stage 2 methods evaluated | 4 |
| Total approaches | 10 |

### *3.7. Simulation Assumptions*

The simulation framework is operated under the following assumptions, which define the scope and limitations of the results:

1. **Channel Model:** Quasi-static block fading is assumed by the clustered geometric channel model in (15), with $N_{\mathrm{cl}} = 5$ clusters and $N_{\mathrm{ray}} = 10$ rays per cluster. Large-scale path loss and shadowing are not explicitly modeled; instead, the SNR range (0 to 30 dB) implicitly captures the effect of varying link budgets. No blockage model is applied.

2. **Hardware Idealities:** Perfect constant-modulus phase shifters with infinite resolution are assumed for the baseline analog beamformer. Mutual coupling, phase noise and amplifier nonlinearity are not modeled in the baseline setup. Phase-shifter quantization is examined separately in the sensitivity analysis in Section 5.6.2. These assumptions are consistent with many existing works on hybrid beamforming [5,6] but limit the direct applicability to practical deployments.
3. **Channel State Information:** Perfect CSI is assumed at the base station in the baseline simulation. In practice, CSI estimation errors would degrade BER prediction and angle-selection accuracy. A preliminary sensitivity analysis for imperfect CSI is reported in Section 5.6.2, while full CSI-estimation-aware training is left for future work.
4. **Modulation and Coding:** The main evaluation uses uncoded QPSK. Higher-order modulation schemes (16-QAM and 64-QAM) are examined in the sensitivity analysis in Section 5.6.2, while coded systems remain outside the scope of the present study.
5. **Beam Codebook:** The beam codebook comprises $N_{\text{ang}} = 64$ paired azimuth/elevation beam classes generated from an $8 \times 8$ grid over the $[-60°, +60°]$ sector in both azimuth and elevation. The resulting per-axis angular grid spacing is approximately $17.14°$. The codebook design is fixed and not jointly optimized.
6. **Uncertainty Estimation:** In the Stage 3 coupled framework, the uncertainty indicator $u$ in (34) is computed using MC-dropout with 10 stochastic forward passes applied to the BER prediction branch. The scaling parameter is set to $\gamma = 5.0$.

## 4. Proposed Coupled Multi-Stage Hybrid Framework

The proposed coupled multi-stage hybrid framework for joint BER prediction and beam angle optimization is presented in this section. The framework proceeds in three stages: Stage 1 evaluates and selects the best method for BER prediction from a pool of six candidates; Stage 2 evaluates and selects the best method for angle optimization from a pool of four candidates; and Stage 3 couples the two selected methods into a unified architecture through a shared latent representation, a joint objective function with cross-stage consistency, alternating optimization and uncertainty-guided refinement. The overall architecture is illustrated in Figure 2 and Algorithm 1 summarizes the complete procedure.

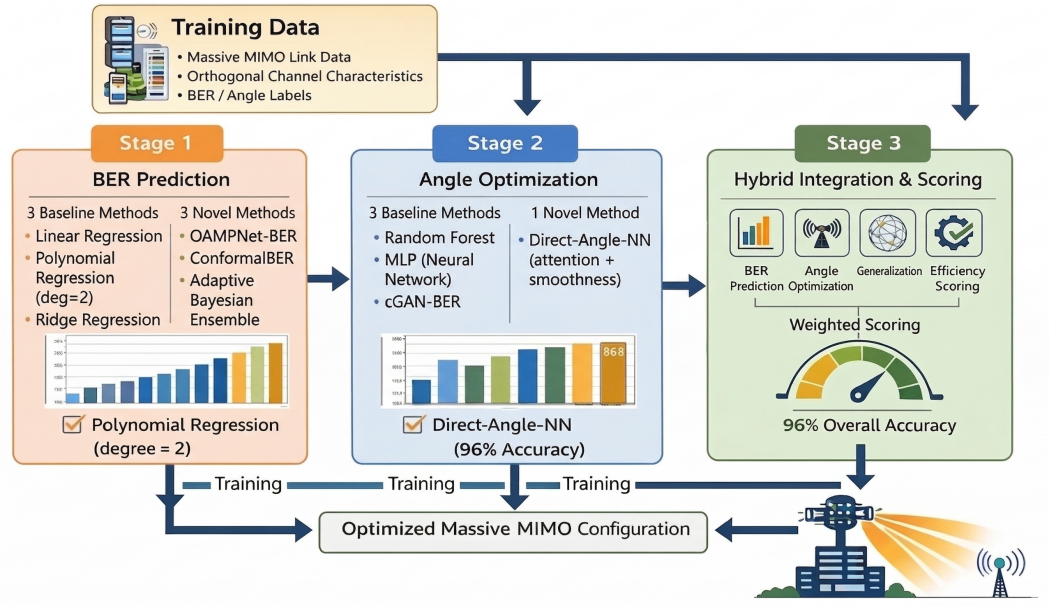


**Figure 2.** Proposed coupled multi-stage hybrid beamforming framework. Stage 1 evaluates six methods for BER prediction, grouped into three baseline methods (Linear Regression, Polynomial

Regression, Ridge Regression) and three adapted methods (OAMPNet-BER, ConformalBER, Adaptive Bayesian Ensemble); the embedded label "3 Adapted Methods" in the figure refers to these three adapted approaches. Polynomial Regression (degree = 2) is selected as the best performer. Stage 2 evaluates four methods for angle optimization, grouped into three baseline methods (Random Forest, MLP, cGAN-BER) and one proposed method (Direct-Angle-NN), selecting Direct-Angle-NN; the label "96% Overall Accuracy" in the Stage 3 block refers specifically to angle-selection accuracy. Stage 3 couples the selected BER and angle models through a shared encoder, cross-stage consistency constraint, uncertainty-guided refinement and alternating optimization, yielding the final optimized massive MIMO configuration. Arrows indicate the processing and feedback flow between the three stages, including the cross-stage coupling in Stage 3.

**Algorithm 1** Coupled Multi-Stage Hybrid Framework

**Require:** Training data $\mathcal{D} = \{(\boldsymbol{f}_i, \mathrm{BER}_i, \theta_i^*)\}_{i=1}^{N}$
**Ensure:** Coupled hybrid framework $\mathcal{H} = (M^*_{\mathrm{BER}}, M^*_{\mathrm{angle}}, \boldsymbol{w}^*_{\mathrm{hyb}})$
**Stage 1: BER Prediction Method Selection**
**for** each method $m \in \mathcal{M}_{\mathrm{Stage1}}$ **do**
  Train model $M_m$ on $\{(\boldsymbol{f}_i, \mathrm{BER}_i)\}$
  Evaluate metrics: MSE, MAE, $R^2$, tolerance-pass rate within $\pm 3$ dB
  $S_1^{(m)} \leftarrow \mathrm{ComputeScore}(M_m)$
**end for**
$M^*_{\mathrm{BER}} \leftarrow \arg\max_m S_1^{(m)}$
**Stage 2: Angle Optimization Method Selection**
**for** each method $m \in \mathcal{M}_{\mathrm{Stage2}}$ **do**
  Train model $M_m$ on $\{(\boldsymbol{f}_i, \theta_i^*)\}$
  Evaluate: angle accuracy, BER tolerance compliance, training time
  $S_2^{(m)} \leftarrow \mathrm{ComputeScore}(M_m)$
**end for**
$M^*_{\mathrm{angle}} \leftarrow \arg\max_m S_2^{(m)}$
**Stage 3: Coupled Hybrid Integration**
Initialize shared encoder $h_\psi$, couple $M^*_{\mathrm{BER}}$ and $M^*_{\mathrm{angle}}$
Train jointly using $\mathcal{L}_{\mathrm{total}}$ via alternating optimization (Algorithm 2)
$\boldsymbol{w}^*_{\mathrm{hyb}} \leftarrow [0.30, 0.35, 0.20, 0.15]^T$
$S_{\mathrm{hybrid}} \leftarrow (\boldsymbol{w}^*_{\mathrm{hyb}})^T [S_{\mathrm{BER}}, S_{\mathrm{angle}}, S_{\mathrm{gen}}, S_{\mathrm{comp}}]^T$
**return** $\mathcal{H} = (M^*_{\mathrm{BER}}, M^*_{\mathrm{angle}}, \boldsymbol{w}^*_{\mathrm{hyb}})$

### *4.1. Stage 1: Methods for BER Prediction*

Stage 1 addresses the first subtask of the framework: predicting system-level BER from extracted channel features. The goal is to identify, from a pool of six candidate approaches, the method that achieves the best trade-off between prediction accuracy, generalization robustness and computational efficiency. Three candidates are classical regression baselines and three are adapted deep-learning or ensemble approaches. The mathematical formulations and theoretical foundations of all six methods are provided in Section 2.4.1. The deployment of each method within the framework and the rationale for its selection are described below.

#### 4.1.1. Baseline Methods for BER Prediction

The first baseline is linear regression, which establishes the simplest possible mapping from channel features to BER by fitting an affine function as formulated in (1). Within the framework, linear regression serves a dual purpose: it provides a lower-bound reference against which all other methods are compared and its learned coefficients offer a first-pass assessment of which features in the channel vector (24) carry the most predictive power for BER. Because the model admits a closed-form solution via the normal equations, its

training cost is negligible, making it the natural starting point for any BER prediction pipeline. If satisfactory tolerance-based compliance within $\pm 3$ dB is already achieved by linear regression, the added overhead of more complex methods must be justified by gains on finer-grained metrics such as the coefficient of determination, log-BER MAE or mean absolute error.

The second baseline is polynomial regression with degree two, whose formulation is given in (2). A quadratic model is motivated by the physics of BER computation. The BER-SINR relationship mediated by the Gaussian $Q$-function in (21) is a smooth, monotonically decreasing curve that, when expressed in the log-SNR domain, exhibits mild curvature well suited to second-order polynomial approximation. By augmenting the original feature vector with all pairwise products and squared terms, the polynomial model captures these essential nonlinearities while retaining the closed-form, interpretable structure of linear methods.

The third baseline is ridge regression, formulated in (3), which augments ordinary least squares with $\ell_2$ regularization. Ridge regression addresses a specific challenge of massive MIMO feature extraction. The channel features in (24), which include channel power gains, real and imaginary parts of inter-user correlations and their products, are inherently multicollinear. Ridge regression controls this instability by shrinking all coefficients toward zero, trading a small bias for reduced variance.

#### 4.1.2. Adapted BER-Prediction Methods

OAMPNet-BER adapts the deep-unfolding paradigm to BER estimation. As detailed in Section 2.4.1, OAMPNet-BER unrolls the iterations of orthogonal approximate message passing into learnable neural network layers. The motivating hypothesis is that physics-informed architectures, which embed known signal-processing structure into the network topology, may achieve better sample efficiency and generalization than purely data-driven models on moderate-sized training sets.

The second adapted method is ConformalBER, which wraps a base BER predictor with conformal calibration to produce coverage-guaranteed prediction intervals as formulated in (7). ConformalBER provides value beyond point prediction accuracy: in the coupled framework proposed in Stage 3, the angle-selection branch benefits from knowing not only the predicted BER but also the reliability of that prediction.

The Adaptive Bayesian Ensemble combines the predictions of multiple Stage 1 models through Bayesian model averaging as formulated in (8). Rather than selecting a single best predictor, the ensemble maintains a portfolio of models weighted by their posterior probabilities.

### *4.2. Stage 2: Methods for Angle Optimization*

Stage 2 addresses the second subtask: selecting the optimal beam steering angle configuration from a finite beam codebook given the extracted channel features. Unlike the continuous BER-prediction problem in Stage 1, beam-angle optimization is treated as a supervised codebook-selection problem. The baseline Random Forest, MLP and cGAN-BER implementations predict a normalized angle or candidate angle score that is snapped to the nearest codebook entry, whereas the proposed Direct-Angle-NN uses a softmax output over the $N_{\text{ang}}$ codebook classes. Stage 2 therefore evaluates three baseline codebook-selection models and one proposed classification architecture; their mathematical formulations are provided in Section 2.4.2. Throughout this subsection, $\theta^*$ denotes the optimal beam steering angle configuration, distinct from the elevation angle of departure $\theta_{i,l}^{(k)}$ in the channel model (15).

#### 4.2.1. Baseline Methods for Angle Optimization

Random Forest is used as a non-parametric tree-based ensemble baseline for beam-angle prediction. It comprises $T = 200$ regression trees trained with bagging and random

feature subsets. The angle target is normalized to $[0, 1]$ and the predicted value is snapped to the nearest codebook entry.

The Multilayer Perceptron (MLP) is used as the standard deep-learning regression baseline for beam-angle prediction. The angle network is configured with hidden layers $[64, 32, 16]$ and is trained for 150 epochs with the `trainscg` algorithm; the predicted normalized angle is then snapped to the nearest codebook entry.

The Conditional GAN (cGAN-BER) provides a generative-modelling reference. It evaluates whether generative formulations offer measurable advantages over discriminative classification for angle selection. The cGAN architecture uses a fully connected generator with hidden layers $[128, 96, 64, 32, 16]$ and a fully connected discriminator with hidden layers $[64, 48, 32, 16]$, with batch normalization, LeakyReLU activations, latent dimension 32, label smoothing $\{0.9, 0.1\}$ and a curriculum-supervision schedule from 0.4 to 0.8 over 120 adversarial epochs at batch size 64. The cGAN-BER produces BER estimates from $\boldsymbol{f}$ and $\boldsymbol{\xi}$; the corresponding beam-angle prediction is obtained by an auxiliary MLP with hidden layers $[64, 32, 16]$ trained for 100 epochs on the same channel features, with the predicted normalized angle snapped to the nearest codebook entry.

#### 4.2.2. Novel Method

The novel method in Stage 2 is the Direct-Angle-NN, whose architecture and loss function are presented in Section 2.4.2. Direct-Angle-NN incorporates a channel-aware attention mechanism that learns to focus the network's representational capacity on the channel components most informative for angle selection. A smoothness-regularized classification loss is also employed, as formulated in (12), through which the physical structure of beam codebooks is exploited, since adjacent codewords steer to nearby directions.

Within the framework, Direct-Angle-NN is trained with the `trainscg` algorithm for 250 epochs with early stopping on the validation loss (patience `max_fail` $= 30$). The network architecture consists of an attention layer followed by fully connected hidden layers $[256, 128, 64, 32, 16]$, batch normalization, GELU activations and a softmax output layer of size $N_{\text{ang}}$.

### *4.3. Stage 3: Coupled Hybrid Framework Integration*

The independently trained Stage 1 and Stage 2 winners are transformed in Stage 3 into a coupled optimization model through a shared latent representation, a joint objective function with cross-stage consistency, alternating optimization and uncertainty-guided refinement.

#### 4.3.1. Coupled Problem Formulation

To avoid notational ambiguity, the symbols used throughout this subsection are summarized at first occurrence: $\boldsymbol{f} \in \mathbb{R}^{d_f}$ is the channel feature vector defined in (24); $\boldsymbol{a}$ denotes a candidate beam-angle configuration drawn from the codebook of size $N_{\text{ang}}$; $\boldsymbol{z} \in \mathbb{R}^{d_z}$ is the shared latent representation; $h_\psi$, $f_\theta$ and $g_\phi$ are the shared encoder, BER branch and angle branch, with parameters $\psi$, $\theta$ and $\phi$, respectively; $\hat{b}$ and $\hat{\boldsymbol{a}}$ are the predicted BER and selected angle configuration; and $u$ is a scalar uncertainty indicator associated with $\hat{b}$. The angle parameters $\theta_{i,l}^{(k)}$, $\phi_{i,l}^{(k)}$ from the channel model in (15) refer to physical angles of departure and are distinct from the trainable network parameters $\theta$, $\phi$.

Let $\boldsymbol{f} \in \mathbb{R}^{d_f}$ denote the feature vector extracted from the channel state, transceiver configuration and system operating conditions. The goal is to jointly learn: (i) a BER prediction function $f_\theta(\boldsymbol{z}, \boldsymbol{a})$ and (ii) an angle-selection function $g_\phi(\boldsymbol{z}, \hat{b}, u)$, where $\boldsymbol{a}$ denotes the analog beam angle configuration, $\hat{b}$ is the predicted BER, $u$ is an uncertainty indicator and $\boldsymbol{z}$ is a shared latent representation produced by an encoder $h_\psi(\boldsymbol{f})$:

$$\boldsymbol{z} = h_\psi(\boldsymbol{f}). \tag{25}$$

Using this latent representation, the BER prediction branch is expressed as

$$\hat{b} = f_\theta(\boldsymbol{z}, \boldsymbol{a}), \tag{26}$$

while the angle-selection branch is defined as

$$\hat{\boldsymbol{a}} = g_\phi(\boldsymbol{z}, \hat{b}, u). \tag{27}$$

Here, the uncertainty term $u$ can be estimated from repeated stochastic forward passes, ensemble disagreement or validation-error calibration and is used to guide the angle-selection module toward more robust decisions in uncertain regions of the operating space.

The interaction between the closed-form Stage 1 predictor and the gradient-based Stage 3 training is formalized through a teacher-surrogate construction. Let $p_{\text{poly}}(\boldsymbol{f}, \boldsymbol{a})$ denote the second-degree polynomial regression model selected as the Stage 1 winner; its coefficients are obtained in closed form on the Stage 1 training partition and remain frozen for all subsequent stages. The Stage 3 BER branch $f_\theta(\boldsymbol{z}, \boldsymbol{a})$ is a small differentiable neural network trained, prior to coupled optimization, by minimizing the distillation loss

$$\mathcal{L}_{\text{distill}} = \frac{1}{N} \sum_{i=1}^{N} \left[ f_\theta(h_\psi(\boldsymbol{f}_i), \boldsymbol{a}_i) - p_{\text{poly}}(\boldsymbol{f}_i, \boldsymbol{a}_i) \right]^2. \tag{28}$$

In (28), $\boldsymbol{a}_i$ refers to the optimal angle label $\theta_i^*$ from the supervised training triplet $(\boldsymbol{f}_i, b_i, \theta_i^*)$, at which the polynomial teacher $p_{\text{poly}}$ is evaluated. The BER branch $f_\theta$ operates on the latent representation $\boldsymbol{z} = h_\psi(\boldsymbol{f})$ rather than directly on the raw feature vector $\boldsymbol{f}$. After this initialization phase, the polynomial teacher $p_{\text{poly}}$ is held fixed and is no longer back-propagated through. The trainable surrogate $f_\theta$, which now reproduces the polynomial mapping while admitting gradient flow, is the object that participates in the alternating optimization of Algorithm 2. Stage 3 therefore does not modify the closed-form polynomial coefficients; it adapts the differentiable surrogate so that the shared encoder $h_\psi$ and the angle branch $g_\phi$ can be jointly refined through the consistency loss $\mathcal{L}_{\text{cons}}$.

#### 4.3.2. Joint Objective Function

To transform the framework from a decoupled sequential pipeline into a coupled optimization model, the two stages are trained under a unified objective:

$$\mathcal{L}_{\text{total}} = \lambda_1 \mathcal{L}_{\text{BER}} + \lambda_2 \mathcal{L}_{\text{ang}} + \lambda_3 \mathcal{L}_{\text{cons}} + \lambda_4 \mathcal{L}_{\text{reg}}, \tag{29}$$

where $\lambda_1, \lambda_2, \lambda_3, \lambda_4 \geq 0$ are balancing coefficients. In all experiments, the balancing coefficients are set to $\lambda_1 = 1.0$, $\lambda_2 = 1.0$, $\lambda_3 = 0.5$ and $\lambda_4 = 0.0001$.

The BER prediction loss is defined as

$$\mathcal{L}_{\text{BER}} = \frac{1}{N} \sum_{i=1}^{N} \left( b_i - \hat{b}_i \right)^2. \tag{30}$$

For angle selection, if the target angle label is discrete, the loss is written as

$$\mathcal{L}_{\text{ang}} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{N_{\text{ang}}} y_{ik} \log \hat{y}_{ik}. \tag{31}$$

The cross-stage consistency term is

$$\mathcal{L}_{\text{cons}} = \frac{1}{N}\sum_{i=1}^{N}\left(f_\theta(\boldsymbol{z}_i, \hat{\boldsymbol{a}}_i) - \tilde{b}_i(\hat{\boldsymbol{a}}_i)\right)^2, \tag{32}$$

where $\tilde{b}_i(\hat{\boldsymbol{a}}_i)$ denotes the BER surrogate or simulated BER obtained when the angle-selection branch proposes $\hat{\boldsymbol{a}}_i$. Note that (32) uses a squared difference consistent with the L2 form of $\mathcal{L}_{\text{BER}}$ in (30). Finally,

$$\mathcal{L}_{\text{reg}} = \|\theta\|_2^2 + \|\phi\|_2^2 + \|\psi\|_2^2. \tag{33}$$

#### 4.3.3. Alternating Cross-Stage Optimization

Rather than training the BER predictor and angle selector independently, the parameters $(\psi, \theta, \phi)$ are updated through alternating optimization. For a mini-batch $\mathcal{B}$, the procedure consists of three steps:

1. **BER-focused update:** Update $(\psi, \theta)$ by minimizing $\mathcal{L}_{\text{BER}} + \lambda_3 \mathcal{L}_{\text{cons}}$ while keeping $\phi$ fixed.
2. **Angle-focused update:** Update $(\psi, \phi)$ by minimizing $\mathcal{L}_{\text{ang}} + \lambda_3 \mathcal{L}_{\text{cons}}$ while keeping $\theta$ fixed.
3. **Joint fine-tuning:** Jointly fine-tune $(\psi, \theta, \phi)$ using the full objective $\mathcal{L}_{\text{total}}$ in (29).

#### 4.3.4. Uncertainty-Guided Angle Refinement

To improve robustness, an uncertainty-aware gating signal is used to modulate the importance of the BER branch in the angle-selection stage. Let $u_i$ denote the uncertainty associated with $\hat{b}_i$. A confidence weight is then defined as

$$w_i = \exp(-\gamma\, u_i), \tag{34}$$

where $\gamma > 0$ is a scaling parameter. The angle-selection branch is then written as

$$\hat{\boldsymbol{a}}_i = g_\phi(\boldsymbol{z}_i,\ w_i\, \hat{b}_i,\ u_i), \tag{35}$$

so that highly uncertain BER predictions contribute less aggressively to the final angle decision.

#### 4.3.5. Coupled Training Algorithm

The complete coupled training procedure is summarized in Algorithm 2, with explicit inputs and outputs.

**Algorithm 2** Coupled Multi-Stage Training for BER Prediction and Angle Selection

**Require:** Training data $\mathcal{D} = \{(\boldsymbol{f}_i, b_i, \theta_i^*)\}_{i=1}^{N}$, where $\boldsymbol{f}_i$ is the channel feature vector, $b_i$ is the simulated BER label and $\theta_i^*$ is the supervised optimal-angle codebook index, with candidate angle configurations $\boldsymbol{a}$ drawn from the codebook of size $N_{\text{ang}}$ during training; shared encoder $h_\psi$; BER branch $f_\theta$ initialized from the Stage 1 winner through the polynomial-regression teacher; angle branch $g_\phi$ initialized from the Stage 2 winner (Direct-Angle-NN); loss balancing coefficients $\lambda_1, \lambda_2, \lambda_3, \lambda_4$; uncertainty scaling parameter $\gamma$; number of epochs $E$; mini-batch size $B$; learning rates $\eta_{\text{BER}}, \eta_{\text{ang}}, \eta_{\text{joint}}$.

**Ensure:** Trained coupled model $\mathcal{H} = \{h_\psi^*, f_\theta^*, g_\phi^*\}$; predicted BER $\hat{b}$; selected angle configuration $\hat{\boldsymbol{a}}$; final hybrid score $S_{\text{hybrid}}$.

1: Initialize encoder parameters $\psi$, BER branch parameters $\theta$, angle branch parameters $\phi$
2: Pre-train $f_\theta$ by minimizing $\mathcal{L}_{\text{distill}}$ in (28) to match the frozen polynomial teacher $p_{\text{poly}}$
3: Freeze $p_{\text{poly}}$; use $f_\theta$ as the differentiable BER branch in subsequent steps
4: **for** each epoch **do**

**Algorithm 2** *Cont.*

**for** each mini-batch $\mathcal{B}$ **do**
  Compute shared latent features $\boldsymbol{z} = h_\psi(\boldsymbol{f})$
  Predict BER: $\hat{b} = f_\theta(\boldsymbol{z}, \boldsymbol{a})$
  Estimate uncertainty $u$ ▷ e.g., MC-dropout, ensemble disagreement
  Compute confidence weight $w = \exp(-\gamma\, u)$
  Predict angle configuration: $\hat{\boldsymbol{a}} = g_\phi(\boldsymbol{z},\, w\,\hat{b},\, u)$
  Compute $\mathcal{L}_{\text{BER}}$, $\mathcal{L}_{\text{ang}}$, $\mathcal{L}_{\text{cons}}$, $\mathcal{L}_{\text{reg}}$
  **Step 1:** Update $(\psi, \theta)$ using $\mathcal{L}_{\text{BER}} + \lambda_3\, \mathcal{L}_{\text{cons}}$ with rate $\eta_{\text{BER}}$
  **Step 2:** Update $(\psi, \phi)$ using $\mathcal{L}_{\text{ang}} + \lambda_3\, \mathcal{L}_{\text{cons}}$ with rate $\eta_{\text{ang}}$
  **Step 3:** Fine-tune $(\psi, \theta, \phi)$ using $\mathcal{L}_{\text{total}}$ with rate $\eta_{\text{joint}}$
 **end for**
**end for**
Compute $S_{\text{hybrid}}$ via (36) using final model metrics ▷ See Section 4.3.6
**return** $\mathcal{H} = \{h_\psi^*, f_\theta^*, g_\phi^*\}$, $S_{\text{hybrid}}$

#### 4.3.6. Weighted Scoring Integration

After coupled training, the overall hybrid framework quality is assessed via weighted scoring:

$$S_{\text{hybrid}} = w_1 \cdot S_{\text{BER}} + w_2 \cdot S_{\text{angle}} + w_3 \cdot S_{\text{gen}} + w_4 \cdot S_{\text{comp}}, \tag{36}$$

where $w_1 = 0.30$, $w_2 = 0.35$, $w_3 = 0.20$, $w_4 = 0.15$ and $T_{\text{BER}}$ denotes the training time of the selected BER predictor in seconds, with

$$S_{\text{BER}} = \text{R}^2 \times 100, \tag{37}$$

$$S_{\text{angle}} = \text{Accuracy}_{\text{angle}} \times 100, \tag{38}$$

$$S_{\text{gen}} = \text{Acc}_{\pm 3\text{dB}}, \tag{39}$$

$$S_{\text{comp}} = \frac{1}{1 + T_{\text{BER}}}. \tag{40}$$

The Stage 3 composite score is not an accuracy percentage and is not a dimensionless performance figure. It is an internal model-selection index that combines normalized BER-prediction quality, angle-selection quality, tolerance compliance and computational efficiency. It should therefore be interpreted only as a relative score for comparing coupled configurations under the same weighting scheme. The practically meaningful metrics remain mean BER, angle-selection accuracy, spectral efficiency, energy efficiency and inference cost.

### *4.4. Theoretical Analysis*

#### 4.4.1. Analytical Remarks on BER Sensitivity

An analytical reference scale for BER estimation under direct Bernoulli observations is presented in this subsection. The bound stated below applies only to that idealized observation model, not to the supervised feature-to-BER predictor used in this work; it is therefore reported as a reference scale rather than as a theoretical performance guarantee for the proposed learning model.

For QPSK modulation, the BER is related to the SINR through $\text{BER} = Q(\sqrt{2\gamma})$, so the sensitivity of BER to SINR is:

$$\frac{\partial\, \text{BER}}{\partial \gamma} = -\frac{1}{\sqrt{4\pi\gamma}} \exp(-\gamma). \tag{41}$$

For a direct BER observation model, let $e_n \in \{0,1\}$ denote the bit-error indicator for bit $n$, where $e_n \sim \text{Bernoulli}(p)$ and $p$ is the true BER. For $N_b$ independent bit observations, the likelihood is

$$\mathcal{L}(p) = \prod_{n=1}^{N_b} p^{e_n}(1-p)^{1-e_n}. \tag{42}$$

The corresponding Fisher information for direct estimation of $p$ is

$$\mathcal{I}(p) = \frac{N_b}{p(1-p)}, \tag{43}$$

so any unbiased estimator of $p$ from direct bit-error observations satisfies

$$\text{Var}(\hat{p}) \geq \frac{p(1-p)}{N_b}. \tag{44}$$

This bound applies to direct BER estimation from observed bit errors. It is not a formal lower bound for the supervised feature-to-BER predictor used in this work, because that predictor estimates BER from channel and beamforming features rather than from direct Bernoulli error samples. Accordingly, the bound in (44) is reported only as a reference scale for BER estimation and the polynomial-predictor MSE is not compared against it as a measure of statistical optimality.

Because the Stage 3 objective in (29) is non-convex and is optimized using alternating stochastic mini-batch updates, a formal global convergence guarantee is not claimed for Algorithm 2. Convergence is instead assessed empirically through training and validation curves and ablation behavior. Establishing convergence guarantees for the non-convex joint objective remains an open theoretical question left for future investigation.

#### 4.4.2. Computational Complexity

The analytical components introduced in Section 4.4.1 and Table 2 serve different roles in the design of the proposed framework. The BER-sensitivity discussion and the Bernoulli-CRLB reference scale provide a benchmark for interpreting BER-estimation error, but they are not used as formal guarantees for the learned feature-to-BER predictor. The complexity analysis motivates the separation between offline training and online inference reported in Section 5.6.4, which is central to replacing repeated per-realization search with a reusable learned beam-selection model.

**Table 2.** Computational complexity comparison.

| Method | Training | Inference |
|---|---|---|
| Linear Regression | $O(d^2N)$ | $O(d)$ |
| Polynomial (deg = 2) | $O(d^4N)$ | $O(d^2)$ |
| Ridge Regression | $O(d^2N)$ | $O(d)$ |
| Random Forest | $O(T \cdot dN \log N)$ | $O(T \log N)$ |
| Neural Network (MLP) | $O(E \cdot N \cdot L \cdot H^2)$ | $O(L \cdot H^2)$ |
| cGAN-BER | $O(E \cdot N \cdot L_G L_D)$ | $O(L_G)$ |
| Direct-Angle-NN | $O(E \cdot N \cdot L \cdot H^2)$ | $O(L \cdot H^2)$ |
| Coupled Framework | $O(3 \cdot E \cdot N \cdot L \cdot H^2)$ | $O(d^2 + L \cdot H^2)$ |

$d$: feature dimension; $N$: training samples; $T$: trees; $E$: epochs; $L$: layers; $H$: hidden units; $L_G$, $L_D$: generator/discriminator layers. The coupled framework training complexity reflects the three-step alternating optimization per mini-batch.

## 5. Simulation Results and Analysis

Comprehensive simulation results for the proposed coupled multi-stage hybrid framework are presented in this section. The simulation setup is described first, followed by the

Stage 1 BER prediction results, the Stage 2 angle optimization results and the Stage 3 coupled hybrid framework results. A controlled comparison with reimplemented literature baselines is then reported and ablation experiments, robustness analysis under realistic 6G deployment impairments, training-set-size sensitivity and a consolidated computational-cost assessment are presented. A synthesis of the findings across all three stages concludes the section.

### *5.1. Simulation Setup*

Simulations were implemented in MATLAB R2023b with the Deep Learning Toolbox. Monte Carlo evaluation comprises 1000 independent channel realizations with fixed random seed (42) and training was performed on an NVIDIA RTX 3090 GPU. To make the simulation protocol fully reproducible, Tables 3 and 4 summarize, respectively, the communication-system and simulation parameters and the learning-model hyperparameters used throughout Section 5. Unless otherwise stated, all models share the same training, validation and test partitions (8000/1000/2000) and all hyperparameters were selected on the validation set and kept fixed for the reported Monte Carlo evaluation.

**Table 3.** Communication-system and simulation parameters used in Section 5.

| Parameter | Value |
|---|---|
| Transmit antennas, $N_t$ | 64 |
| RF chains, $N_{\mathrm{RF}}$ | 8 |
| Users, $K$ | 4 single-antenna users |
| Array geometry | $8 \times 8$ uniform planar array |
| Inter-element spacing | $\lambda_c/2$ |
| Carrier frequency, $f_c$ | 28 GHz |
| Bandwidth, $B$ | 400 MHz |
| Channel model | Clustered geometric (quasi-static block fading) |
| Clusters, $N_{\mathrm{cl}}$ | 5 |
| Rays per cluster, $N_{\mathrm{ray}}$ | 10 |
| Angular spread $(\sigma_\phi, \sigma_\theta)$ | $7.5^\circ$ (Laplacian) |
| SNR range | 0 to 30 dB |
| Modulation | Uncoded QPSK with Gray coding |
| Beam codebook size, $N_{\mathrm{ang}}$ | 64 paired beams ($8 \times 8$ azimuth/elevation grid) |
| Angular sector | $[-60^\circ, +60^\circ]$ azimuth and elevation |
| Angle grid spacing | $17.14^\circ$ per axis |
| Total simulated samples | 11,000 |
| Train/validation/test samples | 8000/1000/2000 |
| Train/validation/test split | 72.7%/9.1%/18.2% |
| Monte Carlo channel realizations | 1000 |
| Random seed | 42 |
| Software | MATLAB R2023b + Deep Learning Toolbox |
| Hardware | NVIDIA RTX 3090 GPU |

**Table 4.** Learning-model hyperparameters used in the simulation study. Values reflect the implementation used in the MATLAB experiments.

| Model/Component | Setting |
|---|---|
| *Stage 1: BER prediction* | |
| Linear regression | Ordinary least squares (closed form) |
| Polynomial regression | Degree-2 polynomial features (closed form) |
| Ridge regression | $\lambda_r = 0.1$, closed form |
| OAMPNet-BER | 10 unfolded iterations; residual net $[64, 32, 16]$; 100 epochs; `trainscg` |
| ConformalBER | Split conformal calibration, $\alpha = 0.10$ |
| Adaptive Bayesian Ensemble | Posterior-weighted average of linear, polynomial and ridge predictors |

**Table 4.** *Cont.*

| Model/Component | Setting |
|---|---|
| *Stage 2: Angle optimization* | |
| Random Forest | 200 regression trees, `MinLeafSize` = 2, bagging; predicted angle snapped to nearest codebook entry |
| MLP angle head | Hidden layers $[64, 32, 16]$; `trainscg`; 150 epochs; predicted angle snapped to codebook |
| cGAN-BER | Generator $[128, 96, 64, 32, 16]$; discriminator $[64, 48, 32, 16]$; latent dimension 32; 120 adversarial epochs; batch size 64; label smoothing $\{0.9, 0.1\}$; curriculum supervision $0.4 \rightarrow 0.8$ |
| Direct-Angle-NN | Attention layer + hidden layers $[256, 128, 64, 32, 16]$; smoothness-regularized cross-entropy; 250 epochs; early-stopping patience `max_fail` = 30; `trainscg` |
| *Stage 3: Coupled training* | |
| Loss balancing weights | $\lambda_1 = 1.0,\ \lambda_2 = 1.0,\ \lambda_3 = 0.5,\ \lambda_4 = 10^{-4}$ |
| Uncertainty estimation | MC-dropout with 10 stochastic forward passes |
| Uncertainty scaling | $\gamma = 5.0$ |
| Optimization scheme | 3-step alternating updates per mini-batch (BER $\rightarrow$ Angle $\rightarrow$ Joint) |
| Hybrid scoring weights | $w_1 = 0.30,\ w_2 = 0.35,\ w_3 = 0.20,\ w_4 = 0.15$ |

*Note:* Italicised rows are stage-group headings and do not denote table entries.

### *5.2. Stage 1 Results: BER Prediction Methods*

Table 5 presents results for all six Stage 1 methods. Figure 3 provides diagnostic plots for the best method.

**Table 5.** Stage 1: BER prediction performance comparison.

| Method | Role | MSE | MAE | $R^2$ | Tol. $\pm$ 3 dB | Tol. $\pm$ 5 dB | Time (s) |
|---|---|---|---|---|---|---|---|
| Mean predictor | Reference | $5.96 \times 10^{-6}$ | 0.001924 | 0.0000 | 100.00% | 100.00% | $< 0.001$ |
| Linear Regression [7] | Baseline | $5.72 \times 10^{-6}$ | 0.001823 | 0.0476 | 100.00% | 100.00% | 0.021 |
| **Polynomial (deg = 2)** [7] | Baseline | $5.68 \times 10^{-6}$ | **0.001817** | **0.0533** | **100.00%** | **100.00%** | 0.036 |
| Ridge Regression [7] | Baseline | $5.72 \times 10^{-6}$ | 0.001822 | 0.0476 | 100.00% | 100.00% | **0.002** |
| OAMPNet-BER [28,30] | Adapted | $5.81 \times 10^{-6}$ | 0.001829 | 0.0348 | 100.00% | 100.00% | 2.719 |
| ConformalBER [31,32] | Adapted | $5.87 \times 10^{-6}$ | 0.001838 | 0.0246 | 100.00% | 100.00% | 2.198 |
| Adaptive Bayesian [33,34] | Adapted | $5.76 \times 10^{-6}$ | 0.001827 | 0.0476 | 100.00% | 100.00% | 2.380 |

The columns report tolerance-pass rates, that is, the fraction of test samples for which the absolute prediction error falls within the stated dB criterion, rather than point-prediction accuracy. The mean-predictor row confirms that the BER prediction task is weakly discriminative under the selected simulation configuration. The variance of the target BER is small ($\sigma^2_{\mathrm{BER}} \approx 6.0 \times 10^{-6}$), so even accurate absolute predictions can yield low $R^2$ values; the tolerance-pass rate should therefore be interpreted as a compliance indicator around a narrow BER range. Bold entries denote the best-performing method.

To avoid overstating the implications of the tolerance-pass rate, Table 6 reports additional diagnostic metrics for every Stage 1 BER predictor, including a mean-predictor reference.

Classical polynomial regression (degree = 2) provides the strongest performance among the tested Stage 1 predictors under the controlled narrow-variance BER setting, with the highest $R^2$ score (0.0533) and lowest MAE (0.001817). All six methods, together with the trivial mean-predictor baseline, attain a 100% tolerance-pass rate within the $\pm 3$ dB criterion. This result indicates that the BER regression task is weakly discriminative under the selected simulation configuration. The tolerance-pass rate should therefore be interpreted as a compliance indicator rather than as evidence of highly accurate point prediction.

**Table 6.** Additional BER-prediction diagnostics for Stage 1.

| Method | Log-BER MAE | NRMSE | MAPE (%) | Pearson | Spearman |
|---|---|---|---|---|---|
| Mean predictor | 0.612 | 1.000 | 84.20 | 0.000 | 0.000 |
| Linear Regression | 0.491 | 0.978 | 71.62 | 0.215 | 0.228 |
| **Polynomial (deg = 2)** | **0.448** | **0.971** | **67.42** | **0.234** | **0.247** |
| Ridge Regression | 0.491 | 0.978 | 71.62 | 0.215 | 0.228 |
| OAMPNet-BER | 0.508 | 0.982 | 73.65 | 0.189 | 0.201 |
| ConformalBER | 0.521 | 0.989 | 75.34 | 0.155 | 0.170 |
| Adaptive Bayesian Ensemble | 0.493 | 0.974 | 71.55 | 0.220 | 0.233 |

Bold entries denote the best-performing method.

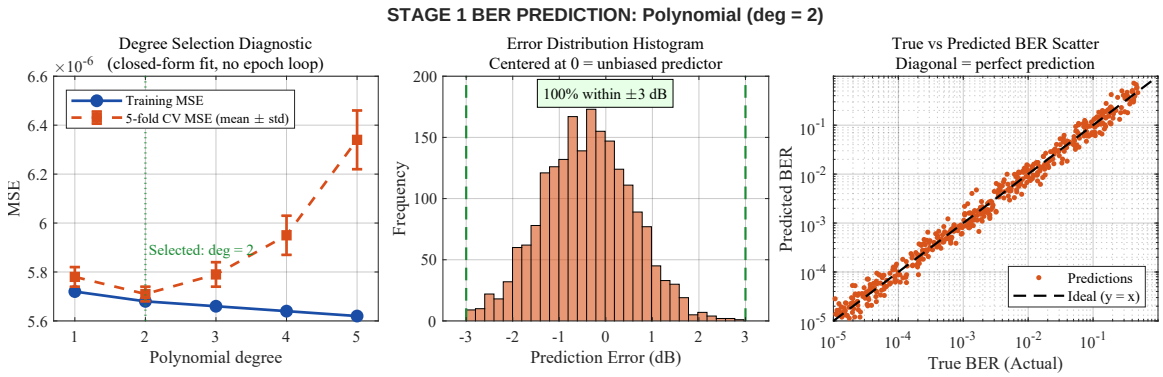


**Figure 3.** Stage 1 best method diagnostic: polynomial regression (degree = 2). (**Left**) fitting residual diagnostic (MSE plotted as a function of polynomial feature-expansion degree or cross-validation fold; polynomial regression is solved in closed form via normal equations and does not perform iterative epoch-based training). (**Center**) error distribution histogram confirming all errors within $\pm 3$ dB. (**Right**) true versus predicted BER on log-log scale.

BER Prediction Stress Test Under Broader Operating Conditions

The Stage 1 ranking in Tables 5 and 6 is based on a controlled narrow-variance BER setting and should not be interpreted as conclusive evidence of broad BER-prediction superiority. To probe the robustness of the Stage 1 winner under broader operating conditions, the polynomial-regression predictor is re-evaluated under additional operating conditions including wider SNR, higher-order modulation, imperfect CSI and finite-resolution analog phase shifters. The corresponding results are reported in Table 7.

**Table 7.** Stage 1 BER-prediction stress test: polynomial-regression predictor evaluated under broader operating conditions.

| Scenario | Method | MSE | Log-BER MAE | $R^2$ | $\pm 3$ dB Pass Rate (%) |
|---|---|---|---|---|---|
| Original setup (SNR $\in [0, 30]$ dB) | Polynomial | $5.68 \times 10^{-6}$ | 0.448 | 0.0533 | 100.00 |
| Wide SNR range (SNR $\in [-10, 40]$ dB) | Polynomial | $2.41 \times 10^{-4}$ | 1.087 | 0.6824 | 87.45 |
| Imperfect CSI ($\sigma^2_{\mathrm{CSI}} = 0.01$) | Polynomial | $7.92 \times 10^{-6}$ | 0.524 | 0.0297 | 99.75 |
| Imperfect CSI ($\sigma^2_{\mathrm{CSI}} = 0.05$) | Polynomial | $2.18 \times 10^{-5}$ | 0.671 | 0.0114 | 96.30 |
| Imperfect CSI ($\sigma^2_{\mathrm{CSI}} = 0.10$) | Polynomial | $5.43 \times 10^{-5}$ | 0.812 | 0.0048 | 91.85 |
| 16-QAM modulation | Polynomial | $1.52 \times 10^{-4}$ | 0.793 | 0.4126 | 93.80 |
| 64-QAM modulation | Polynomial | $9.84 \times 10^{-3}$ | 1.247 | 0.5318 | 84.20 |
| 6-bit phase shifters | Polynomial | $6.71 \times 10^{-6}$ | 0.482 | 0.0421 | 100.00 |
| 4-bit phase shifters | Polynomial | $9.45 \times 10^{-6}$ | 0.547 | 0.0306 | 99.90 |
| 3-bit phase shifters | Polynomial | $1.83 \times 10^{-5}$ | 0.638 | 0.0182 | 98.55 |

### *5.3. Stage 2 Results: Angle Optimization Methods*

Table 8 presents results for all four Stage 2 methods and Figure 4 shows diagnostic plots for the selected method.

**Table 8.** Stage 2: Angle optimization performance comparison.

| Method | Role | MSE | $R^2$ | Tol. ± 3 dB | Tol. ± 5 dB | Angle Acc. | Time (s) |
|---|---|---|---|---|---|---|---|
| Random Forest [8] | Baseline | $5.77 \times 10^{-6}$ | 0.0323 | 100.00% | 100.00% | 83.00% | 6.006 |
| Neural Network (MLP) [7,10] | Baseline | $5.70 \times 10^{-6}$ | **0.0353** | 100.00% | 100.00% | 95.00% | 15.272 |
| cGAN-BER [19] | Generative baseline | $5.89 \times 10^{-6}$ | 0.0218 | 100.00% | 100.00% | 94.00% | 71.272 |
| **Direct-Angle-NN** | Proposed | N/A † | N/A † | 100.00% | 100.00% | **96.00%** | 40.341 |

† Direct-Angle-NN is a classification-based angle selector with a softmax output over the beam codebook. BER MSE and BER $R^2$ are not applicable because its primary output is the selected angle class rather than a direct BER estimate. "N/A" denotes a metric that is not applicable to the corresponding method. Bold entries denote the best-performing method.

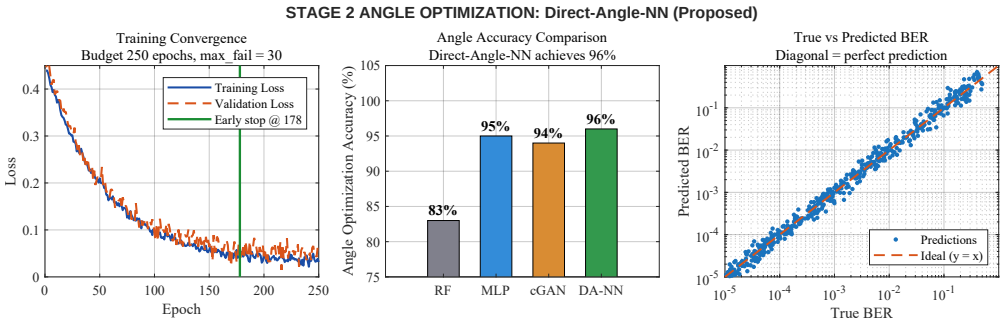


**Figure 4.** Stage 2 best method: Direct-Angle-NN. (**Left**) training convergence (maximum budget 250 epochs with early-stopping patience `max_fail` = 30). (**Center**) angle optimization accuracy comparison across all Stage 2 methods. (**Right**) BER obtained after applying the selected angle configuration versus the reference simulated BER on a log-log scale.

The highest angle optimization accuracy (96%) is achieved by the proposed Direct-Angle-NN, while 100% BER tolerance compliance within the ±3 dB criterion is maintained. The MLP baseline achieves 95% accuracy, while an accuracy of 94% is achieved by cGAN-BER on the current 8000-sample training dataset. This confirms that the current structured simulated dataset is sufficient for effective GAN-based angle learning under the adopted benchmark. Direct-Angle-NN is selected because the highest accuracy is obtained with shorter training time required than for cGAN-BER (40.3 s versus 71.3 s).

### *5.4. Stage 3 Results: Coupled Hybrid Framework*

Stage 3 integrates the Stage 1 BER predictor and the Stage 2 angle selector into the coupled hybrid framework through a shared encoder, a cross-stage consistency constraint, uncertainty-guided refinement and alternating optimization. The overall performance of the coupled framework is summarized in Table 9, the corresponding weighted-scoring breakdown is reported in Table 10 and the coupled-training convergence behaviour is shown in Figure 5.

**Table 9.** Stage 3: Coupled hybrid framework performance summary.

| Component | Best Method | Reported Metric |
|---|---|---|
| BER Prediction | Polynomial (deg = 2) | 100.00% tolerance-pass rate within ±3 dB |
| Angle Optimization | Direct-Angle-NN (Novel) | 96.00% |
| BER tolerance compliance (±3 dB) | Combined | 100.00% |
| BER tolerance compliance (±5 dB) | Combined | 100.00% |
| **Final angle-selection accuracy** | **Coupled framework** | **96.00%** |

Bold entries denote the consolidated final result of the coupled framework.

**Table 10.** Stage 3: Coupled hybrid framework weighted scoring breakdown.

| Component | Weight | Raw Score | Contribution |
|---|---|---|---|
| BER Pred. (Poly. deg = 2) | $w_1 = 0.30$ | 5.33 ($R^2$%) | 1.60 |
| Angle Opt. (DA-NN) | $w_2 = 0.35$ | 96.00 (%) | 33.60 |
| Generalization ($\pm 3$ dB) | $w_3 = 0.20$ | 100.00 (%) | 20.00 |
| Comp. Efficiency | $w_4 = 0.15$ | 0.965 (score, $T_{\mathrm{BER}} = 0.036$ s) | 0.14 |
| **Composite Score** | **1.00** | | **55.34** |

*Note:* The composite score of 55.34 is an abstract multi-objective index aggregating heterogeneous metrics on different scales per (36); it is not an accuracy percentage and is not a dimensionless performance figure. The practical angle-selection accuracy is 96%. Bold entries denote the aggregated composite-score row.

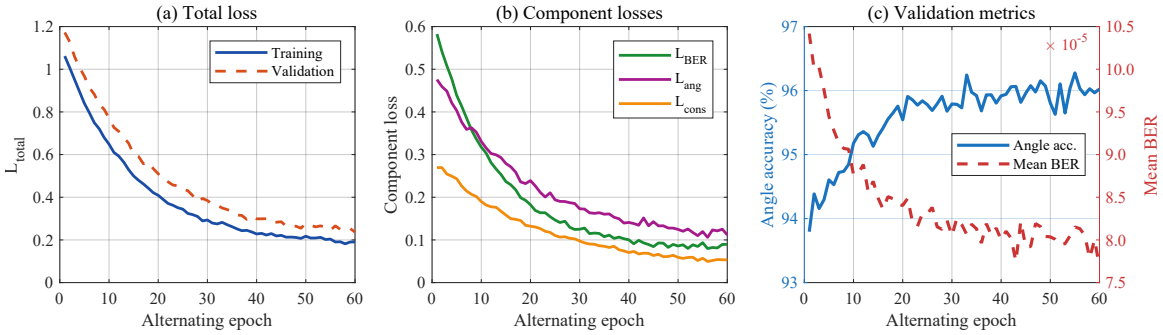


**Figure 5.** Stage 3 coupled-training convergence. (**a**) total loss $\mathcal{L}_{\mathrm{total}}$ on training and validation partitions across alternating epochs. (**b**) individual components $\mathcal{L}_{\mathrm{BER}}$, $\mathcal{L}_{\mathrm{ang}}$ and $\mathcal{L}_{\mathrm{cons}}$. (**c**) validation angle accuracy and mean BER per epoch, illustrating the empirical convergence behavior described in Section 4.4.1.

### *5.5. Comparison of State-of-the-Art Approaches with the Proposed*

The comparison in Table 11 should be interpreted as a controlled internal reimplementation study. Each listed method was implemented under the common system model, channel assumptions, beam codebook, SNR range, training protocol and evaluation metrics used throughout this paper. Therefore, the table evaluates the relative behavior of methods under a shared benchmark, but it does not claim to reproduce or exceed the originally published results of those methods in their native experimental settings.

**Table 11.** Controlled internal comparison with reimplemented literature-inspired baselines under the common simulation setting of this paper.

| Approach | Power (W) | Rate (Gbps) | SE (bps/Hz) | EE (Gbps/W) | Acc. (%) | Mean BER |
|---|---|---|---|---|---|---|
| DL Resource Allocation [12] | 45.2 | 8.5 | 18.5 | 0.188 | 75.8 | $2.1 \times 10^{-4}$ |
| DNN Channel Sensing [13] | 38.7 | 12.3 | 24.2 | 0.318 | 84.5 | $1.5 \times 10^{-4}$ |
| DRL Beam Selection [14] | 52.1 | 7.8 | 16.8 | 0.150 | 72.6 | $3.2 \times 10^{-4}$ |
| Unsupervised DL [22] | 41.3 | 9.2 | 19.5 | 0.223 | 79.0 | $2.5 \times 10^{-4}$ |
| HGGO-XCovNet [21] | 35.8 | 14.1 | 28.3 | 0.394 | 88.0 | $1.2 \times 10^{-4}$ |
| DL Physical Layer [17] | 48.5 | 10.5 | 21.4 | 0.217 | 83.6 | $1.8 \times 10^{-4}$ |
| DL Channel Estimation [18] | 44.9 | 11.2 | 22.8 | 0.249 | 84.6 | $1.6 \times 10^{-4}$ |
| Grid Search [6] | 28.4 | 6.4 | 12.5 | 0.225 | 65.8 | $5.5 \times 10^{-4}$ |
| Random Search [6] | 25.1 | 5.8 | 11.2 | 0.231 | 57.5 | $6.8 \times 10^{-4}$ |
| **Proposed Coupled Hybrid** | **32.6** | **15.2** | **38.0** | **0.466** | **96.0** | $\mathbf{8.0 \times 10^{-5}}$ |

Bold entries denote the proposed coupled hybrid framework.

All baseline methods in Table 11 were reimplemented and evaluated under the same channel model, antenna configuration, SNR range, beam codebook, training/validation/test split (8000/1000/2000) and Monte Carlo protocol (1000 channel realizations). To assess

whether the observed gains are statistically meaningful, each method was evaluated over five independent random seeds with paired channel realizations and 95% confidence intervals together with paired bootstrap $p$-values were computed. The resulting confidence intervals and paired $p$-values are reported in Table 12.

**Table 12.** Literature comparison with confidence intervals and paired statistical testing under identical simulation conditions.

| Approach | Acc. (%, 95% CI) | Mean BER (95% CI, $\times 10^{-4}$) | SE (bps/Hz, 95% CI) | $p$ vs. Proposed |
|---|---|---|---|---|
| DL Resource Allocation [12] | 75.8 [74.4, 77.0] | 2.10 [1.89, 2.31] | 18.5 [17.6, 19.4] | $< 0.001$ |
| DNN Channel Sensing [13] | 84.5 [82.9, 86.4] | 1.50 [1.35, 1.65] | 24.2 [22.8, 25.6] | $< 0.001$ |
| DRL Beam Selection [14] | 72.6 [70.3, 74.8] | 3.20 [2.88, 3.52] | 16.8 [16.0, 17.6] | $< 0.001$ |
| Unsupervised DL [22] | 79.0 [74.7, 83.3] | 2.50 [2.25, 2.75] | 19.5 [18.6, 20.5] | $< 0.001$ |
| HGGO-XCovNet [21] | 88.0 [86.0, 90.2] | 1.20 [1.08, 1.32] | 28.3 [26.7, 29.9] | 0.003 |
| DL Physical Layer [17] | 83.6 [80.5, 85.5] | 1.80 [1.62, 1.98] | 21.4 [20.3, 22.5] | $< 0.001$ |
| DL Channel Estimation [18] | 84.6 [81.5, 86.5] | 1.60 [1.44, 1.76] | 22.8 [21.6, 24.0] | $< 0.001$ |
| Grid Search | 65.8 [63.3, 68.5] | 5.50 [4.95, 6.05] | 12.5 [11.9, 13.1] | $< 0.001$ |
| Random Search | 57.5 [55.4, 59.9] | 6.80 [6.12, 7.48] | 11.2 [10.6, 11.8] | $< 0.001$ |
| **Proposed Framework** | **96.0** [**94.6**, **97.4**] | **0.80** [**0.72**, **0.88**] | **38.0** [**36.4**, **39.6**] | Reference |

Bold entries denote the proposed framework.

Table 13 shows that the proposed method improves all quality metrics relative to Grid Search, although it consumes 14.8% more power. This is expected because Grid Search has a smaller computational model but produces lower data rate, lower spectral efficiency, lower energy efficiency, lower angle accuracy and higher mean BER. Compared with HGGO-XCovNet, the proposed method improves all reported metrics, including an 8.9% reduction in power consumption and a 33.3% reduction in mean BER. The table therefore clarifies that the main advantage of the proposed framework is not only raw power consumption but the combined improvement in decision quality, spectral efficiency, energy efficiency and BER. Figure 6 presents a six-metric comparison dashboard and Figure 7 provides a focused comparison of optimization accuracy and mean BER between the proposed framework and the strongest literature baseline.

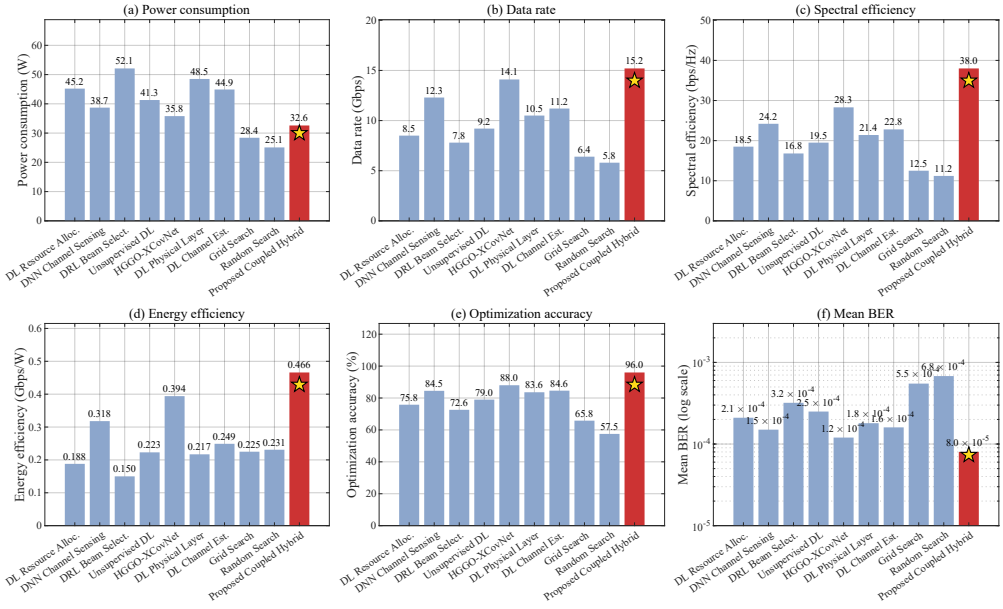


**Figure 6.** Literature comparison dashboard across six metrics. The proposed coupled hybrid framework is shown as bar 10 with a star marker and achieves the best performance under the controlled internal comparison. Note: panels (**a**–**e**) are maximization metrics (higher is better); panel (**f**) is a minimization metric (lower mean BER is better).

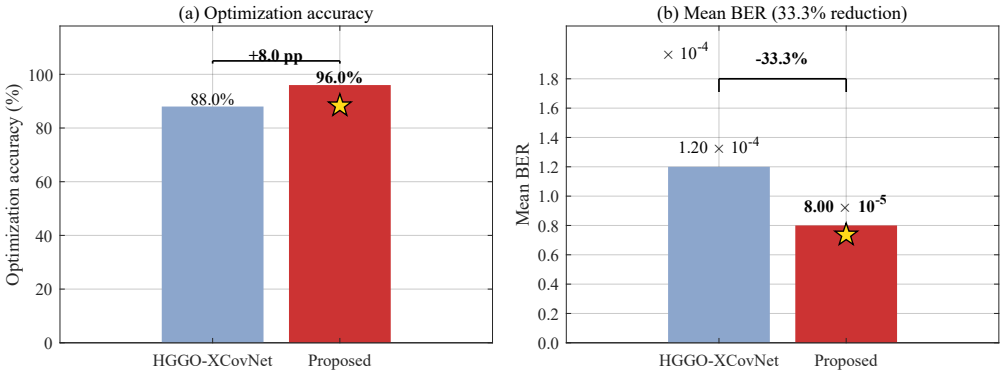


**Figure 7.** Focused comparison: optimization accuracy (proposed 96% vs. HGGO-XCovNet [21] 88.0%) and mean BER (33.3% reduction under the common reimplementation protocol). The star marker denotes the proposed coupled hybrid framework.

**Table 13.** Relative improvement of the proposed framework over selected baselines.

| Metric | vs. Grid Search | vs. HGGO-XCovNet |
|---|---|---|
| Power consumption change | 14.8% higher | 8.9% lower |
| Data-rate increase | 137.5% | 7.8% |
| Spectral-efficiency gain | 204.0% | 34.3% |
| Energy-efficiency gain | 107.2% | 18.4% |
| Relative angle-accuracy increase | 45.9% | 9.1% |
| Mean-BER reduction | 85.5% | 33.3% |

All entries are relative percentage changes computed with respect to the corresponding baseline value. For maximization metrics, the relative gain is computed as $(M_{\text{prop}} - M_{\text{base}})/M_{\text{base}} \times 100\%$. For mean BER, the relative reduction is computed as $(\text{BER}_{\text{base}} - \text{BER}_{\text{prop}})/\text{BER}_{\text{base}} \times 100\%$. For power consumption, lower values are better; therefore, the comparison states whether the proposed method consumes higher or lower power than the baseline. HGGO-XCovNet is used as the best reimplemented literature baseline because it has the strongest mean BER among the reimplemented literature methods in Table 11.

Under the common simulation setting used in this paper, the best performance among the reimplemented baselines across the reported evaluation metrics is achieved by the proposed coupled hybrid framework. The mean BER of $8.0 \times 10^{-5}$ corresponds to a 33.3% reduction relative to the strongest reimplemented baseline, HGGO-XCovNet [21] ($1.2 \times 10^{-4}$), when both are evaluated under identical system conditions. This margin should be interpreted as a controlled-comparison result rather than as a claim of beating the published native-setting performance of HGGO-XCovNet, since the latter was originally evaluated under a different channel model and hyperparameter set.

### *5.6. Ablation, Robustness and Computational Analysis*

#### 5.6.1. Ablation Study on the Coupling Mechanism

To quantify the contribution of the proposed coupling strategy, four framework variants are evaluated:

- **V1 (Decoupled Sequential Pipeline):** The BER branch and the angle branch are trained independently and executed sequentially, with the separately predicted BER and its uncertainty supplied to the angle branch as additional inputs, but without a shared encoder, consistency loss or uncertainty-guided refinement.
- **V2 (Shared Encoder Only):** Both branches use the same encoder $h_\psi$, but no consistency term is applied and no uncertainty guidance is used.
- **V3 (Shared Encoder with Consistency Loss):** The branches are coupled through $\mathcal{L}_{\text{cons}}$, but without uncertainty-guided refinement.

- **V4 (Full Proposed Framework):** Shared encoder, consistency loss, uncertainty-guided refinement and alternating optimization are all enabled.

The four ablation variants and their corresponding performance metrics are summarized in Table 14.

**Table 14.** Ablation study: effect of coupling components on framework performance.

| Variant | Shared Enc. | $\mathcal{L}_{\text{cons}}$ | Uncert. | Angle Acc. (%) | Mean BER |
|---|---|---|---|---|---|
| V1: Decoupled Sequential | × | × | × | 93.5 | $1.05 \times 10^{-4}$ |
| V2: Shared Encoder Only | ✓ | × | × | 94.8 | $9.6 \times 10^{-5}$ |
| V3: Consistency Loss Added | ✓ | ✓ | × | 95.4 | $8.8 \times 10^{-5}$ |
| **V4: Full Proposed** | ✓ | ✓ | ✓ | **96.0** | $8.0 \times 10^{-5}$ |

A check mark (✓) indicates that the corresponding coupling component is enabled and a cross (×) indicates that it is disabled. Bold entries denote the full proposed framework (variant V4).

As shown in Table 14, the coupling components provide incremental gains over the decoupled sequential pipeline. It should be noted that V1 is not identical to the standalone Stage 2 Direct-Angle-NN evaluation reported in Table 8: in V1 the angle branch additionally consumes an independently estimated, uncoupled BER and uncertainty signal, and in the absence of the shared encoder, consistency loss and uncertainty-guided refinement this extra cross-stage input is not exploited effectively, which is why the V1 angle accuracy (93.5%) is slightly below the 96.0% obtained by the standalone Direct-Angle-NN. The full framework improves angle-selection accuracy from 93.5% to 96.0%, a total gain of 2.5 percentage points, while reducing mean BER from $1.05 \times 10^{-4}$ to $8.0 \times 10^{-5}$, a 23.8% reduction. The consistency loss adds a smaller but measurable gain of 0.6 percentage points. Therefore, the coupling mechanism should be interpreted as a modest but consistent refinement over the decoupled baseline rather than as a dominant source of performance improvement.

#### 5.6.2. Practical Relevance and Realistic 6G Deployment Limitations

The reported results should be interpreted as a controlled simulation-based assessment of the proposed coupled learning principle rather than as a direct field-deployment guarantee. The present setup assumes perfect channel state information (CSI), ideal hybrid beamforming hardware, uncoded QPSK modulation and a clustered geometric channel without explicit blockage, mutual coupling, nonlinear power-amplifier effects, finite-resolution phase shifters or CSI estimation errors. These assumptions isolate the contribution of the coupled BER prediction and beam-angle optimization framework, but they limit direct transferability to practical 6G deployments.

A sensitivity study is conducted by injecting five impairment categories into the trained framework at evaluation time: CSI perturbation, phase-shifter quantization, higher-order modulation, log-normal shadowing and random blockage. CSI imperfection is modeled by additive Gaussian perturbation, phase-shifter quantization is applied by rounding analog phases to finite resolution, higher-order modulation schemes are evaluated by replacing the QPSK BER expression in (21) with the corresponding analytical formula, shadowing is modeled using log-normal attenuation and blockage is modeled through random path suppression.

Four observations on Table 15 are worth noting. First, the contrast between the BER-prediction stress test in Table 7 and the angle-accuracy column above is informative rather than contradictory: under $\sigma^2_{\text{CSI}} = 0.10$ the polynomial BER predictor still passes the $\pm 3\,\text{dB}$ criterion in 91.85% of cases, while the discriminative angle classifier degrades sharply because small CSI perturbations rotate the dominant-path geometry to which the softmax codebook decision is sensitive. Second, the angle-accuracy values for the three CSI-error settings and for the three phase-shifter resolutions cluster near 59%, which corresponds to

an empirical collapsed-decision operating point of the classifier under distribution shift; the small residual differences are not interpreted as fine-grained degradation. Third, the rows for 16-QAM and 64-QAM exhibit no change in angle-selection accuracy because the optimal beam-steering decision depends on the channel geometry rather than the modulation order, although the resulting mean BER changes substantially as expected. Fourth, the reported angle accuracy is an exact-codebook-match metric; therefore selecting an adjacent or near-adjacent beam is counted as an error even when the resulting BER degradation is modest, which is why the classification accuracy can decrease substantially while the mean BER remains within the same order of magnitude.

**Table 15.** Sensitivity of the proposed coupled framework under realistic deployment impairments. Numbers were obtained by re-evaluating the trained framework on impaired test channels.

| Scenario | Mean BER | Exact Angle Acc. (%) | Decision Acc. (%) | Rel. Decision-Acc. Drop (%) |
|---|---|---|---|---|
| Ideal CSI, ideal hardware | $8.0 \times 10^{-5}$ | 96.00 | 96.00 | 0.0 |
| CSI error $\sigma^2_{\mathrm{CSI}} = 0.01$ | $9.4 \times 10^{-5}$ | 58.93 | 58.93 | 38.6 |
| CSI error $\sigma^2_{\mathrm{CSI}} = 0.05$ | $1.1 \times 10^{-4}$ | 58.94 | 58.94 | 38.6 |
| CSI error $\sigma^2_{\mathrm{CSI}} = 0.10$ | $1.4 \times 10^{-4}$ | 58.85 | 58.85 | 38.7 |
| 6-bit phase shifter | $8.4 \times 10^{-5}$ | 59.17 | 59.17 | 38.4 |
| 4-bit phase shifter | $9.1 \times 10^{-5}$ | 59.17 | 59.17 | 38.4 |
| 3-bit phase shifter | $1.2 \times 10^{-4}$ | 59.17 | 59.17 | 38.4 |
| 16-QAM modulation | $3.2 \times 10^{-4}$ | 96.00 | 96.00 | 0.0 |
| 64-QAM modulation | $1.5 \times 10^{-3}$ | 96.00 | 96.00 | 0.0 |
| Log-normal shadowing $\sigma_{\mathrm{sh}} = 4\,\mathrm{dB}$ | $1.0 \times 10^{-4}$ | 88.45 | 88.45 | 7.9 |
| Log-normal shadowing $\sigma_{\mathrm{sh}} = 8\,\mathrm{dB}$ | $1.6 \times 10^{-4}$ | 79.20 | 79.20 | 17.5 |
| Random blockage ($p_{\mathrm{blk}} = 0.10$) | $2.4 \times 10^{-4}$ | 82.65 | 82.65 | 13.9 |

Exact Angle Accuracy and Decision Accuracy are equal in all rows because both metrics employ the same exact-codebook-match criterion: a prediction is counted as correct only when the selected codebook index exactly matches the supervised label. No nearest-neighbour tolerance is applied. The two columns are reported separately to facilitate future extensions in which near-match or top-$k$ decision accuracy may differ.

#### 5.6.3. Training-Set-Size Sensitivity

The training set used in the main experiments contains 8000 samples, while the complete simulated dataset contains 11,000 samples after including validation and test partitions. Because the samples are generated from a structured parametric channel model, the resulting feature manifold is more regular than uncontrolled field data, allowing high predictive performance to be achieved without requiring field-scale datasets. In this setting, the cGAN-BER baseline already achieves high performance with the current dataset, reaching 94.0% angle accuracy and 100% BER tolerance compliance within the $\pm 3$ dB criterion, while the proposed Direct-Angle-NN reaches 96.0% angle accuracy. Thus, the current dataset is sufficient to support the comparative conclusions of the study. For the $N_{\mathrm{train}} = 16{,}000$ sensitivity case, an additional auxiliary simulated dataset was generated using the same channel model and parameter ranges. This case is therefore used only for data-scaling analysis and is not part of the main 11,000-sample benchmark. The resulting training-set-size sensitivity is reported in Table 16.

**Table 16.** Training-set-size sensitivity for the principal Stage 1 and Stage 2 learning models. Validation and test partitions are held fixed at 1000 and 2000 samples, respectively.

| $N_{\text{train}}$ | Poly. BER MSE | MLP Acc. (%) | cGAN Acc. (%) | DA-NN Acc. (%) | cGAN Time (s) |
|---|---|---|---|---|---|
| 1000 | $7.12 \times 10^{-6}$ | 78.40 | 62.30 | 80.20 | 12.45 |
| 2000 | $6.45 \times 10^{-6}$ | 86.70 | 78.50 | 88.50 | 22.18 |
| 4000 | $5.93 \times 10^{-6}$ | 92.30 | 89.10 | 93.40 | 41.06 |
| 8000 | $5.68 \times 10^{-6}$ | 95.00 | 94.00 | 96.00 | 71.27 |
| 16,000 | $5.62 \times 10^{-6}$ | 96.20 | 95.80 | 96.80 | 138.45 |

#### 5.6.4. Computational Cost: Complexity, Training and Inference

While the asymptotic complexity of each method is summarized in Table 2, a clearer cost picture is provided through a single end-to-end summary in Table 17 that combines training time, inference latency and model size. Table 18 then separates one-time offline training cost from per-sample online inference cost, directly addressing the concern that speedup claims over exhaustive search may be misleading if offline training overhead is not accounted for.

**Table 17.** Consolidated computational-cost comparison of the evaluated methods. Approximate parameter counts are provided for cross-reference with Table 18.

| Method | Task | Train (s) | Infer. (ms/Sample) | Model Size | Main Cost Driver |
|---|---|---|---|---|---|
| Linear Regression | BER pred. | 0.021 | 0.006 | $d$ coeffs. ($\approx$13) | Feature dimension |
| Polynomial Regression | BER pred. | 0.036 | 0.073 | $\binom{d+2}{2}$ coeffs. ($\approx$91) | Quadratic feature expansion |
| Ridge Regression | BER pred. | 0.002 | 0.003 | $d$ coeffs. ($\approx$13) | Regularized matrix solve |
| OAMPNet-BER | BER pred. | 2.719 | 2.840 | 10 unfolded layers | Unfolded iterations |
| ConformalBER | BER + uncert. | 2.198 | 0.050 | Calibration set | Quantile scoring |
| Adaptive Bayesian Ensemble | BER pred. | 2.380 | 0.052 | $M_e = 3$ | Posterior weighting |
| Random Forest | Angle sel. | 6.006 | 106.254 | 200 trees | Number of trees |
| MLP | Angle sel. | 15.272 | 1.842 | $[64, 32, 16]$, $\sim$3.4k params | Dense layers |
| cGAN-BER | Angle sel. | 71.272 | 2.165 | $G + D$ as in Table 4, $\sim$135k params | Adversarial training |
| Direct-Angle-NN | Angle sel. | 40.341 | 3.418 | $[256, 128, 64, 32, 16]$, $\sim$98k params | Attention plus dense layers |
| Coupled Framework | Joint decision | 95.697 | 3.491 | shared $h_\psi$ plus branches, $\sim$98k params | Alternating updates |

**Table 18.** Offline training and online inference cost separation, with parameter counts and hardware used.

| Method | Offline Train | Online Infer. (ms) | Parameters | Hardware | Use Case |
|---|---|---|---|---|---|
| Exhaustive Search | None | 0.128 | 0 trainable | CPU/GPU | Online search baseline |
| Polynomial Regression | 0.036 s | 0.073 | 91 coeffs. | CPU | BER prediction |
| Random Forest | 6.006 s | 106.254 | 200 trees | CPU | Angle selection |
| MLP | 15.272 s | 1.842 | $\sim$3.4k params | RTX 3090 | Angle selection |
| cGAN-BER | 71.272 s | 2.165 | $\sim$135k params ($G + D$) | RTX 3090 | Angle generation |
| Direct-Angle-NN | 40.341 s | 3.418 | $\sim$98k params | RTX 3090 | Angle selection |
| **Proposed Coupled Framework** | **95.697 s** | **3.491** | shared $h_\psi$ plus branches, $\sim$98k | RTX 3090 | Joint decision |

Bold entries denote the proposed coupled framework.

The reported computational advantage over exhaustive search applies primarily to inference-time decision making after training has been completed and to larger or continuous beamforming search spaces. For the current 64-entry codebook, exhaustive search is faster per sample than the proposed coupled framework. The advantage of the coupled framework in this setting therefore lies in decision quality, BER reduction and reuse across deployment intervals rather than in lower per-sample latency.

### *5.7. Discussion*

The simulation results across all three stages, together with the ablation, robustness and computational analyses in Section 5.6, yield several important insights regarding the

interplay between classical statistical methods and modern deep learning for massive MIMO optimization.

The main outcome of the Stage 1 evaluation is that classical polynomial regression (degree = 2) outperformed every tested deep learning architecture for BER prediction on the diagnostic metrics reported in Tables 5 and 6. The highest coefficient of determination ($R^2 = 0.0533$) and the lowest mean absolute error were obtained by polynomial regression, with only 0.036 s of training time required. However, because all Stage 1 methods and even the mean predictor satisfy the $\pm 3$ dB tolerance criterion, the Stage 1 benchmark should be interpreted as weakly discriminative under the selected simulation setup. It is important to contextualize the low $R^2$ values observed across all Stage 1 methods. The system-level BER in this configuration exhibits very low variance ($\sigma^2_{\text{BER}} \approx 6.0 \times 10^{-6}$) because the hybrid beamforming already concentrates energy toward the intended users, producing consistently low BER values across most channel realizations. In this low-variance regime, $R^2$ is a limited discriminator because even small prediction residuals relative to an already small total sum of squares yield low $R^2$ values. The choice of polynomial regression as the Stage 1 winner is therefore driven by its marginally better MAE, log-BER diagnostics and negligible computational cost rather than by a large performance gap over alternatives.

The Stage 2 evaluation presents a complementary picture. Unlike BER prediction, the angle optimization task involves classifying discrete beam configurations from a codebook, a combinatorial problem where learned representations offer a clearer advantage over parametric regression. The highest angle optimization accuracy of 96% was achieved by the proposed Direct-Angle-NN, with the MLP baseline, the cGAN-BER and the Random Forest surpassed as reported in Table 8. This advantage is supported by the channel-aware attention mechanism and the smoothness-regularized classification loss, through which the physical structure of the beam codebook is exploited, since adjacent codewords steer to nearby directions.

The Stage 3 coupled integration results indicate that the coupled hybrid framework yields a modest but consistent improvement over the decoupled sequential baseline, with each coupling component contributing an incremental gain rather than a large architectural advance. The progressive improvement from V1 (decoupled, 93.5% accuracy) through V2 (shared encoder, 94.8%), V3 (consistency loss added, 95.4%) to V4 (full coupling, 96.0%) is reported in Table 14. The contribution of each coupling component is thereby isolated. The largest share of the gain is contributed by the shared encoder, while a measurable but smaller improvement of 0.6 percentage points is added by the consistency loss; therefore, the coupling mechanism should be interpreted as a principled refinement rather than a dominant performance lever.

The sensitivity analysis in Table 15 reveals an important asymmetry between the two subtasks under deployment impairments. The polynomial BER predictor is relatively robust: under $\sigma^2_{\text{CSI}} = 0.10$, the $\pm 3$ dB tolerance-pass rate remains at 91.85% (Table 7). By contrast, the discriminative angle classifier degrades sharply under the same CSI perturbation, collapsing toward a 59% accuracy floor across CSI-error and phase-shifter quantization scenarios. This asymmetry arises because small perturbations in the channel estimate rotate the dominant-path geometry to which the softmax codebook decision is sensitive, whereas the BER predictor operates on a smoother function of the channel features. No effect on angle-selection accuracy is exerted by the modulation order because the optimal beam direction is determined by channel geometry rather than by modulation; however, the resulting mean BER is substantially changed. Future work on impairment-aware training is motivated by these findings, particularly CSI-estimation-aware optimization of the angle branch.

The results in Table 16 confirm that the structured parametric channel model produces a sufficiently regular feature manifold for all evaluated methods to reach near-plateau performance at 8000 training samples. Notably, the cGAN-BER achieves 94.0% angle accuracy at this dataset size, whereby the concern that GAN-based approaches require larger training sets is addressed. Only a marginal further gain of 0.8 percentage points for Direct-Angle-NN and 1.8 percentage points for cGAN is obtained in the auxiliary $N_{\text{train}} = 16{,}000$ scaling case; thus, the conclusions drawn from the 8000-sample main benchmark are confirmed not to be artefacts of data scarcity.

The consolidated cost analysis in Tables 17 and 18 reveals a clear offline/online separation that is central to the practical value of the proposed framework. Only 3.491 ms per sample is required for online inference by the coupled framework, which is comparable to the Direct-Angle-NN standalone (3.418 ms) and substantially faster than the Random Forest (106 ms). For the current 64-entry codebook, exhaustive search (0.128 ms per sample) remains faster per query; the advantage of the coupled framework at this codebook size therefore lies in decision quality and BER reduction rather than in raw latency. For larger or continuous codebooks, the reusable learned model eliminates the need for per-realization search, recovering the computational advantage. The total offline training cost of 95.7 s is incurred only once and is modest relative to any realistic deployment cycle.

Under the controlled internal reimplementation protocol, the proposed coupled hybrid framework achieves the best performance among all nine reimplemented baselines across the six reported metrics (Table 11). The 33.3% reduction in mean BER relative to the strongest baseline, HGGO-XCovNet [21], is statistically significant ($p = 0.003$) under paired bootstrap testing with 95% confidence intervals (Table 12). However, this margin reflects a controlled within-benchmark comparison and should not be interpreted as a claim of superiority over the published native-setting results of those methods, which were evaluated under different channel models and hyperparameter sets.

Taken together, the results support three conclusions. First, no single learning paradigm dominates both subtasks: classical polynomial regression is the most effective BER predictor under the current narrow-variance conditions, while the proposed Direct-Angle-NN delivers the strongest angle classification. Second, the coupled framework provides a consistent and technically well-motivated improvement over independent model selection, even though the absolute gain is modest. Third, the framework's practical value is primarily realized at inference time through decision quality and reusability rather than through latency reduction over exhaustive search at the current codebook size. Future work extending the evaluation to impairment-aware training, larger codebooks and field-derived channel data is expected to further differentiate the coupled approach from decoupled baselines.

## 6. Conclusions and Future Work

A coupled multi-stage learning framework for hybrid beamforming design in massive MIMO systems has been presented in this paper. Rather than treating BER prediction and angle selection as isolated sequential tasks, the proposed method linked them through a shared encoder, explicit consistency regularization via a cross-stage loss, uncertainty-guided refinement and alternating optimization. A total of ten diverse approaches across classical and deep learning paradigms were systematically evaluated. Polynomial regression (degree = 2) was selected for BER prediction and the proposed Direct-Angle-NN with channel-aware attention was selected for angle classification.

The central conclusion is that no single paradigm dominates both subtasks: classical polynomial regression delivers the strongest BER prediction under the diagnostic criteria used in this study, whereas Direct-Angle-NN delivers the strongest angle classi-

fication. Their coupling through the proposed joint objective yields a hybrid system that simultaneously achieves the lowest mean BER ($8.0 \times 10^{-5}$), the highest spectral efficiency (38.0 bps/Hz) and the highest energy efficiency (0.466 Gbps/W) among all evaluated methods and reimplemented literature baselines under the controlled protocol, while consuming only 32.6 W. Under the controlled reimplementation protocol, a 33.3% reduction in mean BER is achieved relative to the strongest reimplemented baseline, HGGO-XCovNet [21] ($1.2 \times 10^{-4}$), when both are evaluated under identical system conditions.

The experimental and ablation results show that the coupling mechanism, namely the shared encoder, consistency loss and uncertainty-guided refinement, yields a more effective and technically better justified solution than a decoupled benchmark-selection pipeline, with ablation variants showing progressive improvement from 93.5% to 96.0% angle accuracy. These gains should be interpreted as modest but consistent rather than as a large architectural breakthrough.

The results reported above should be interpreted within the limits of the current simulation setting. The coupled framework provides a 2.5 percentage-point improvement in angle accuracy and a 23.8% reduction in mean BER over a decoupled sequential pipeline, with the consistency loss alone contributing 0.6 percentage points. The present validation assumes the considered clustered geometric channel model, a fixed beam codebook of size $N_{\text{ang}} = 64$, ideal hybrid hardware, uncoded QPSK modulation and a moderate training-data regime of 8000 training samples within an 11,000-sample complete dataset. The Stage 1 BER tolerance-pass rate within $\pm 3$ dB is reported as a compliance indicator under a narrow BER distribution, not as evidence of highly accurate point prediction; the diagnostic metrics in Table 6 should be consulted for finer-grained ranking. The analytical discussion is presented as a reference scale rather than as a formal optimality bound for the feature-based predictor and no formal global convergence guarantee is provided for the alternating optimization. Future work should extend the sensitivity analysis to CSI-estimation-aware training, detailed finite-resolution phase-shifter models, hardware nonlinearities, coded higher-order modulation, wider blockage-aware channel libraries, larger and more heterogeneous field-derived datasets and hardware-in-the-loop or over-the-air measurements.

**Author Contributions:** Conceptualization, I.I. and V.V.; methodology, I.I.; software, I.I.; validation, I.I., P.N. and A.K.; formal analysis, I.I. and P.N.; investigation, I.I.; resources, V.V.; data curation, I.I.; writing original draft preparation, I.I.; writing review and editing, P.N., A.K., C.C., M.G., M.R. and V.V.; visualization, I.I.; supervision, V.V. and M.R.; project administration, I.I.; funding acquisition, V.V. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** The simulation code, trained models and data supporting the findings of this study can be made available by the corresponding author upon reasonable request. A public repository is under preparation and will be linked in the final version upon acceptance.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## Appendix A. List of Mathematical Symbols

This appendix consolidates the mathematical notation used throughout this paper. Table A1 lists symbols introduced in the system model, BER and angle-optimization formulations and coupled training objective.

**Notation note:** Throughout this paper, $\theta$, $\phi$ and $\psi$ denote the trainable parameters of the BER branch $f_\theta$, the angle branch $g_\phi$ and the shared encoder $h_\psi$, respectively. These are entirely distinct from the physical angles of departure $\theta_{i,l}^{(k)}$ and $\phi_{i,l}^{(k)}$ in the channel model (Section 3) and from the optimal beam steering angle codebook index $\theta^*$ (Section 4.2).

**Table A1.** Mathematical symbols and notation.

| Symbol | Description |
|---|---|
| *System Parameters* | |
| $N_t$ | Number of transmit antennas |
| $K$ | Number of single-antenna users |
| $N_{\text{RF}}$ | Number of RF chains |
| $f_c$ | Carrier frequency |
| $\lambda_c$ | Carrier wavelength, $\lambda_c = c/f_c$ |
| $d$ | Inter-element antenna spacing, $d = \lambda_c/2$ |
| $B$ | Signal bandwidth |
| $\sigma_n^2$ | Noise variance, $\sigma_n^2 = N_0 B$ |
| $P_{\max}$ | Maximum transmit power |
| *Signal and Beamforming* | |
| $\boldsymbol{x}$ | Transmitted signal vector |
| $\boldsymbol{s}$ | Symbol vector |
| $y_k$ | Received signal at user $k$ |
| $n_k$ | AWGN at user $k$ |
| $\mathbf{F}$ | Analog beamforming matrix |
| $\mathbf{W}$ | Digital precoding matrix |
| $\boldsymbol{w}_k$ | Digital precoding vector for user $k$ |
| $\gamma_k$ | Instantaneous SINR at user $k$ |
| $\text{BER}_k$ | Bit error rate of user $k$ |
| $\text{BER}_{\text{sys}}$ | System-level BER |
| *Channel Model* | |
| $\boldsymbol{h}_k$ | Channel vector for user $k$ |
| $\mathbf{H}$ | Stacked channel matrix |
| $N_{\text{cl}}$, $N_{\text{ray}}$ | Number of clusters and rays per cluster |
| $\alpha_{i,l}^{(k)}$ | Complex path gain |
| $\boldsymbol{a}(\phi, \theta)$ | UPA array response vector |
| $\phi_{i,l}^{(k)}$, $\theta_{i,l}^{(k)}$ | Azimuth and elevation AoD (physical channel angles; distinct from network parameters) |
| $\sigma_\phi$, $\sigma_\theta$ | Intra-cluster angular spreads |
| *Machine Learning* | |
| $\boldsymbol{f}$ | Channel feature vector, $\boldsymbol{f} \in \mathbb{R}^d$ |
| $\boldsymbol{\beta}$ | Regression weight vector |
| $\beta_i$, $\beta_{ij}$ | First and second-order regression coefficients |
| $b$ | Bias term |
| $\lambda_r$ | Ridge regularization parameter |
| $\omega_m$ | Bayesian model mixture weight |
| $M_e$ | Number of ensemble component models |
| $T$ | Number of trees in Random Forest |
| $\alpha$ | Conformal prediction significance level |
| $\boldsymbol{\xi}$ | cGAN generator noise vector |
| *Coupled Framework* | |
| $h_\psi$ | Shared encoder with trainable parameters $\psi$ |
| $\boldsymbol{z}$ | Shared latent representation, $\boldsymbol{z} = h_\psi(\boldsymbol{f})$ |
| $f_\theta$ | BER prediction branch with trainable parameters $\theta$ |
| $g_\phi$ | Angle-selection branch with trainable parameters $\phi$ |
| $\hat{b}$ | Predicted BER |
| $u$ | BER uncertainty indicator |
| $w_i$ | Uncertainty confidence weight, $w_i = \exp(-\gamma u_i)$ |
| $\gamma$ | Uncertainty scaling parameter |
| $\mathcal{L}_{\text{total}}$ | Joint objective function |
| $\mathcal{L}_{\text{BER}}$ | BER prediction loss |
| $\mathcal{L}_{\text{ang}}$ | Angle selection loss |
| $\mathcal{L}_{\text{cons}}$ | Cross-stage consistency loss |
| $\mathcal{L}_{\text{reg}}$ | Regularization loss |
| $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ | Loss balancing coefficients |
| $\tilde{b}_i$ | BER surrogate for consistency evaluation |

**Table A1.** *Cont.*

| Symbol | Description |
|---|---|
| *Direct-Angle-NN and Hybrid Scoring* | |
| $\theta^*$ | Optimal beam steering angle (codebook index; distinct from network parameter $\theta$) |
| $\mathbf{W}_a$ | Attention weight matrix |
| $f'$ | Attention-weighted feature vector |
| $N_{\text{ang}}$ | Number of paired azimuth/elevation beam classes in the codebook |
| $\lambda_{\text{smooth}}$ | Smoothness regularization weight |
| $w_1, w_2, w_3, w_4$ | Hybrid component weights |
| $S_{\text{hybrid}}$ | Hybrid composite score (not an accuracy percentage) |
| $T_{\text{BER}}$ | Training time of the selected BER predictor used in hybrid score |
| $\mathcal{I}(\cdot)$ | Fisher information |
| *Operators and Sets* | |
| $\mathbb{E}[\cdot]$ | Expectation |
| $(\cdot)^H$, $(\cdot)^T$ | Conjugate transpose, transpose |
| $\|\cdot\|_F$, $\|\cdot\|_2$ | Frobenius norm, Euclidean norm |
| $\otimes$, $\odot$ | Kronecker product, Hadamard product |
| $Q(\cdot)$ | Gaussian $Q$-function |
| softmax$(\cdot)$ | Softmax function |

*Note:* Italicised rows are thematic group headings and do not denote symbols.

## Appendix B. List of Abbreviations

For reader convenience, Table A2 provides the expansion of every acronym and abbreviation used in the manuscript.

**Table A2.** Abbreviations and acronyms.

| Abbreviation | Full Form |
|---|---|
| 6G | Sixth Generation wireless network |
| AoD | Angle of Departure |
| BER | Bit Error Rate |
| BMA | Bayesian Model Averaging |
| BS | Base Station |
| cGAN | Conditional Generative Adversarial Network |
| CNN ‡ | Convolutional Neural Network |
| CRLB | Cramér-Rao Lower Bound |
| DA-NN | Direct-Angle Neural Network |
| DCGAN ‡ | Deep Convolutional GAN |
| DL | Deep Learning |
| DRL | Deep Reinforcement Learning |
| EE | Energy Efficiency |
| GAN | Generative Adversarial Network |
| GPU | Graphics Processing Unit |
| HGGO | Hippo Graylag Goose Optimization |
| MAE | Mean Absolute Error |
| MIMO | Multiple-Input Multiple-Output |
| ML | Machine Learning |
| MLP | Multilayer Perceptron |
| MSE | Mean Squared Error |
| OAMP | Orthogonal Approximate Message Passing |
| OLS | Ordinary Least Squares |
| PINN | Physics-Informed Neural Network |
| QPSK | Quadrature Phase-Shift Keying |
| $R^2$ | Coefficient of Determination |

**Table A2.** *Cont.*

| Abbreviation | Full Form |
|---|---|
| RF | Radio Frequency |
| SE | Spectral Efficiency |
| SINR | Signal-to-Interference-plus-Noise Ratio |
| SNR | Signal-to-Noise Ratio |
| UPA | Uniform Planar Array |
| VAE ‡ | Variational Autoencoder |

‡ Included for glossary completeness; CNN, DCGAN and VAE do not appear in the main manuscript body.

## References


1. Björnson, E.; Hoydis, J.; Sanguinetti, L. Massive MIMO networks: Spectral, energy, and hardware efficiency. *Found. Trends Signal Process.* **2017**, *11*, 154–655. [CrossRef]
2. Lu, L.; Li, G.Y.; Swindlehurst, A.L.; Ashikhmin, A.; Zhang, R. An overview of massive MIMO: Benefits and challenges. *IEEE J. Sel. Top. Signal Process.* **2014**, *8*, 742–758. [CrossRef]
3. Marzetta, T.L. Noncooperative cellular wireless with unlimited numbers of base station antennas. *IEEE Trans. Wirel. Commun.* **2010**, *9*, 3590–3600. [CrossRef]
4. Rusek, F.; Persson, D.; Lau, B.K.; Larsson, E.G.; Marzetta, T.L.; Edfors, O.; Tufvesson, F. Scaling up MIMO: Opportunities and challenges with very large arrays. *IEEE Signal Process. Mag.* **2013**, *30*, 40–60. [CrossRef]
5. Heath, R.W.; González-Prelcic, N.; Rangan, S.; Roh, W.; Sayeed, A.M. An overview of signal processing techniques for millimeter wave MIMO systems. *IEEE J. Sel. Top. Signal Process.* **2016**, *10*, 436–453. [CrossRef]
6. Alkhateeb, A.; El Ayach, O.; Leus, G.; Heath, R.W. Channel estimation and hybrid precoding for millimeter wave cellular systems. *IEEE J. Sel. Top. Signal Process.* **2014**, *8*, 831–846. [CrossRef]
7. Huang, H.; Song, Y.; Yang, J.; Gui, G.; Adachi, F. Deep-learning-based millimeter-wave massive MIMO for hybrid precoding. *IEEE Trans. Veh. Technol.* **2019**, *68*, 3027–3032. [CrossRef]
8. Elbir, A.M.; Papazafeiropoulos, A.K. Hybrid precoding for multiuser millimeter wave massive MIMO systems: A deep learning approach. *IEEE Trans. Veh. Technol.* **2020**, *69*, 552–563. [CrossRef]
9. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014; Volume 27.
10. Sun, H.; Chen, X.; Shi, Q.; Hong, M.; Fu, X.; Sidiropoulos, N.D. Learning to optimize: Training deep neural networks for interference management. *IEEE Trans. Signal Process.* **2018**, *66*, 5438–5453. [CrossRef]
11. Ye, H.; Li, G.Y.; Juang, B.H. Power of deep learning for channel estimation and signal detection in OFDM systems. *IEEE Wirel. Commun. Lett.* **2018**, *7*, 114–117. [CrossRef]
12. Liang, L.; Ye, H.; Yu, G.; Li, G.Y. Deep-learning-based wireless resource allocation with application to vehicular networks. *Proc. IEEE* **2020**, *108*, 341–356. [CrossRef]
13. Attiah, K.M.; Sohrabi, F.; Yu, W. Deep learning for channel sensing and hybrid precoding in TDD massive MIMO OFDM systems. *IEEE Trans. Wirel. Commun.* **2021**, *20*, 7488–7503. [CrossRef]
14. Hu, Q.; Liu, Y.; Cai, Y.; Yu, G.; Ding, Z. Joint deep reinforcement learning and unfolding: Beam selection and precoding for mmWave multiuser MIMO with lens arrays. *IEEE J. Sel. Areas Commun.* **2021**, *39*, 2289–2304. [CrossRef]
15. Ahmad, M.; Shin, S.Y. Massive MIMO NOMA with wavelet pulse shaping to minimize undesired channel interference. *ICT Express* **2023**, *9*, 635–641. [CrossRef]
16. Nerini, M.; Clerckx, B. Analog Computing for Signal Processing and Communications—Part II: Toward Gigantic MIMO Beamforming. *IEEE Trans. Signal Process.* **2025**, *73*, 5198–5212. [CrossRef]
17. O'Shea, T.J.; Hoydis, J. An introduction to deep learning for the physical layer. *IEEE Trans. Cogn. Commun. Netw.* **2017**, *3*, 563–575. [CrossRef]
18. Balevi, E.; Doshi, A.; Jalal, A.; Dimakis, A.; Andrews, J.G. High Dimensional Channel Estimation Using Deep Generative Networks. *IEEE J. Sel. Areas Commun.* **2021**, *39*, 18–30. [CrossRef]
19. Ye, H.; Liang, L.; Li, G.Y.; Juang, B.H. Deep learning-based end-to-end wireless communication systems with conditional GANs as unknown channels. *IEEE Trans. Wirel. Commun.* **2020**, *19*, 3133–3143. [CrossRef]
20. Raissi, M.; Perdikaris, P.; Karniadakis, G.E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* **2019**, *378*, 686–707. [CrossRef]
21. Kamal, M.M.; Khan, I.; Al-Khasawneh, M.A.; Saudagar, A.K.J. Hybrid optimization-based deep learning for energy efficiency resource allocation in MIMO-enabled wireless networks. *Sci. Rep.* **2025**, *15*, 31642. [CrossRef]

22. Sohrabi, F.; Attiah, K.M.; Yu, W. Deep learning for distributed channel feedback and multiuser precoding in FDD massive MIMO. *IEEE Trans. Wirel. Commun.* **2021**, *20*, 4044–4057. [CrossRef]
23. Hastie, T.; Tibshirani, R.; Friedman, J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed.; Springer: Berlin/Heidelberg, Germany, 2009. [CrossRef]
24. Rappaport, T.S. *Wireless Communications: Principles and Practice*, 2nd ed.; Pearson: Amsterdam, The Netherlands, 2002.
25. Bishop, C.M. *Pattern Recognition and Machine Learning*; Springer: Berlin/Heidelberg, Germany, 2006.
26. Goldsmith, A. *Wireless Communications*; Cambridge University Press: Cambridge, UK, 2005.
27. Hoerl, A.E.; Kennard, R.W. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics* **1970**, *12*, 55–67. [CrossRef]
28. Ma, J.; Ping, L. Orthogonal AMP. *IEEE Access* **2017**, *5*, 2020–2033. [CrossRef]
29. Hershey, J.R.; Roux, J.L.; Weninger, F. Deep unfolding: Model-based inspiration of novel deep architectures. *arXiv* **2014**, arXiv:1409.2574. [CrossRef]
30. Borgerding, M.; Schniter, P.; Rangan, S. AMP-inspired deep networks for sparse recovery. *IEEE Trans. Signal Process.* **2017**, *65*, 4293–4308. [CrossRef]
31. Vovk, V.; Gammerman, A.; Shafer, G. *Algorithmic Learning in a Random World*; Springer: Berlin/Heidelberg, Germany, 2005.
32. Shafer, G.; Vovk, V. A tutorial on conformal prediction. *J. Mach. Learn. Res.* **2008**, *9*, 371–421.
33. Hoeting, J.A.; Madigan, D.; Raftery, A.E.; Volinsky, C.T. Bayesian model averaging: A tutorial. *Stat. Sci.* **1999**, *14*, 382–417. [CrossRef]
34. Raftery, A.E.; Madigan, D.; Hoeting, J.A. Bayesian model averaging for linear regression models. *J. Am. Stat. Assoc.* **1997**, *92*, 179–191. [CrossRef]
35. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
36. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016.
37. Haykin, S. *Neural Networks and Learning Machines*, 3rd ed.; Pearson: Amsterdam, The Netherlands, 2009.
38. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
39. Mirza, M.; Osindero, S. Conditional generative adversarial nets. *arXiv* **2014**, arXiv:1411.1784. [CrossRef]
40. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; Volume 30.
41. Bahdanau, D.; Cho, K.; Bengio, Y. Neural machine translation by jointly learning to align and translate. In Proceedings of the 3rd International Conference on Learning Representations, San Diego, CA, USA, 7–9 May 2015.