Deep Unfolding with Kernel-based Quantization in MIMO Detection

Zeyi Ren [1] **Jingreng Lei** [1] **Yichen Jin** [1] **Ermo Hua** [2] **Qingfeng Lin** [1] **Chen Zhang** [1] **Bowen Zhou** [2 3] **Yik-Chung Wu** [1]

Abstract

The development of edge computing places critical demands on energy-efficient model deployment for multiple-input multiple-output (MIMO) detection tasks. Deploying deep unfolding models such as PGD-Nets and ADMM-Nets into resource-constrained edge devices using quantization methods is challenging. Existing quantization methods based on quantization aware training (QAT) suffer from performance degradation due to their reliance on parametric distribution assumption of activations and static quantization step sizes. To address these challenges, this paper proposes a novel kernel-based adaptive quantization (KAQ) framework for deep unfolding networks. By utilizing a joint kernel density estimation (KDE) and maximum mean discrepancy (MMD) approach to align activation distributions between full-precision and quantized models, the need for prior distribution assumptions is eliminated. Additionally, a dynamic step size updating method is introduced to adjust the quantization step size based on the channel conditions of wireless networks. Extensive simulations demonstrate that the accuracy of proposed KAQ framework outperforms traditional methods and successfully reduces the model's inference latency.

1. Introduction

In the realm of wireless communications, the proliferation of edge computing presents new opportunities and challenges for efficient data processing (Park et al., 2019; Elbamby et al., 2019; Hu et al., 2020). Edge computing allows for local data processing on devices such as user equipment or small base stations (BS), reducing latency and enhancing privacy, which is particularly crucial for real-time applications like mobile communications (Zhao et al., 2025; Bai et al., 2020; Jeong et al., 2018). A key task in these systems is multiple-input multiple-output (MIMO) detection (Hu et al., 2023; Wang et al., 2025; Shao & Ma, 2021), which requires accurate and swift estimation of transmitted symbols from received signals.

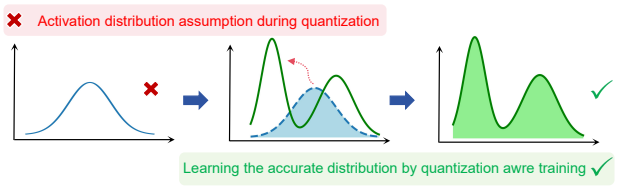

Figure 1. The process of increasing the awareness of accurate activation distribution for models

Mathematically, two categories of optimization algorithms have been extensively applied to MIMO detection tasks and demonstrated to be highly effective: proximal gradient descent (PGD) algorithm (Lin et al., 2024b; Ren et al., 2025) and alternating direction method of multipliers (ADMM) (Chang et al., 2016; Ye et al., 2020) algorithm. PGD algorithm effectively handles non-smooth regularizers in MIMO detection by iteratively optimizing differentiable and proximal terms (Li et al., 2019), while ADMM decouples complex constraints through variable splitting and dual updates, ensuring convergence under non-convex signal recovery settings, making them suitable for MIMO detection tasks.

However, despite their theoretical merits, both PGD and ADMM algorithms face practical challenges in latency-sensitive BS deployments due to their iterative nature and computational overhead (Yu et al., 2023; Popovski et al., 2018). Proximal methods require multiple iterations to converge under non-smooth regularizers, while ADMM's variable splitting and dual updates introduce coordination delays. These limitations challenge their direct applicability for practical adoption.

On the other hand, deep learning-based algorithms have become popular in communication research due to their low inference latency (Barbarossa et al., 2023; Yilmaz et al.,

[1]Department of Electrical and Electronic Engineering, The University of Hong Kong, HKSAR, China [2]Tsinghua University [3]Shanghai AI Laboratory. Correspondence to: Yik-Chung Wu <ycwu@eee.hku.hk>.

Proceedings of the 42^{nd} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025.

2024; Xu et al., 2023; Lei et al., 2024; Shang et al., 2019). However, black box deep learning designs (e.g., multi-layer perceptrons (MLP) or convolutional neural networks (Lei et al., 2022)) do not incorporate the domain knowledge of communication systems. This usually results in unsatisfactory performance or the neural network not being able to converge during training.To address this, deep unfolding techniques have been employed to design efficient detection models (Han et al., 2025; Lin et al., 2024a), which transform iterative optimization algorithms into neural networks by regarding each iteration as one layer of the neural networks, offer a model-based deep learning paradigm for wireless communication tasks.

From the above arguments, applying deep unfolding models like PGD-Net (Mou et al., 2022) and ADMM-Net (Johnston et al., 2021) to the MIMO detection problem seems to be an obvious choice. However, these models typically employ full-precision floating-point operations, making it difficult to deploy on resource-constrained edge devices (Elgabli et al., 2020; Han et al., 2016).

To address this, quantization is a promising way to reduce the complexity of model activations by converting high-bit floating-point data into lower-bit integer representations, preserving model functionality while minimizing computational overhead.

Quantization methods can be broadly categorized into two types, post-training quantization (PTQ) (Xiao et al., 2023; Huang et al., 2025; 2024) and quantization-aware training (QAT) (Chen et al., 2024). PTQ quantizes a pre-trained model without further training, using a calibration dataset to determine quantization parameters. Although straightforward and computationally friendly for large models, PTQ often results in significant accuracy loss, particularly for small models like deep unfolding models. QAT, on the other hand, integrates quantization into the training process, using 'fake quantization' layers to simulate quantization effects (Jacob et al., 2018; Wu et al., 2023). This allows the model to learn representations robust to quantization errors, generally leading to better accuracy than PTQ for deep unfolding (Khobahi et al., 2021). However, traditional QAT methods rely on parametric distribution assumptions when designing the loss functions (Gong et al., 2019), resulting in performance degradation, especially when the actual data distributions are nonparametric or vary across different parts of the network.

To address this limitation, this paper introduces a novel kernel-based adaptive quantization (KAQ) framework specifically designed for deep unfolding networks such as PGD-Net and ADMM-Net. KAQ is a quantization-aware training framework that leverages a joint kernel density estimation (KDE) (Scardapane et al., 2015; Li et al., 2016) and maximum mean discrepancy (MMD) (Tolstikhin et al., 2016) approach to align the activation distributions of the full-precision and quantized models during training. This approach maps the low-dimensional quantization discrepancies into a high-dimensional reproducing kernel Hilbert space (RKHS) to enable gradient-driven optimization by leveraging the kernel-induced feature representation (Zhang et al., 2023b; 2024; Hua et al., 2025). By doing so, KAQ eliminates the need for prior distribution assumptions (Zhang et al., 2023a; 2025), leading to more accurate quantization.

Additionally, KAQ incorporates a dynamic step size updating that adjusts the quantization step size based on the signal-to-noise ratio (SNR). This adaptability ensures that the model is optimally quantized according to the channel condition of the communication system, minimizing information loss.

In the following Section 2 and Section 3, derivation and theoretical analysis focus on implementing the KAQ framework on the PGD-Net, while in Section 4. we also experimentally demonstrate the ADMM-Net quantization using KAQ framework.

Extensive simulations demonstrate that the proposed KAQ framework outperforms traditional quantization methods in terms of performance while successfully reducing the model's inference latency. This advancement paves the way for efficient and accurate deployment of deep unfolding networks on resource-constrained edge devices.

2. System Model and Problem Formulation

2.1. System Model

We consider a multiple-input multiple-output (MIMO) communication system with N_t transmit antennas and N_r receive antennas. The received signal vector $\mathbf{y} \in \mathbb{C}^{N_r}$ is modeled as

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

where $\mathbf{x} \in \mathbb{C}^{N_t}$ denotes the transmitted signal vector, $\mathbf{H} \in \mathbb{C}^{N_r \times N_t}$ represents the channel matrix, and $\mathbf{n} \in \mathbb{C}^{N_r}$ is the additive white Gaussian noise (AWGN) vector with zero mean and covariance $\sigma^2\mathbf{I}$.

The entries of $\mathbf{x}$ are assumed to be drawn from a finite modulation alphabet $\mathcal{X}$, such as quadrature amplitude modulation (QAM). To enable the application of deep unfolding and the corresponding quantization techniques, we convert the complex-valued system into its real-valued equivalent. Specifically, we define

$$\tilde{\mathbf{y}} = \begin{bmatrix} \mathrm{Re}(\mathbf{y}) \\ \mathrm{Im}(\mathbf{y}) \end{bmatrix}, \quad \tilde{\mathbf{x}} = \begin{bmatrix} \mathrm{Re}(\mathbf{x}) \\ \mathrm{Im}(\mathbf{x}) \end{bmatrix},$$

$$\tilde{\mathbf{H}} = \begin{bmatrix} \mathrm{Re}(\mathbf{H}) & -\mathrm{Im}(\mathbf{H}) \\ \mathrm{Im}(\mathbf{H}) & \mathrm{Re}(\mathbf{H}) \end{bmatrix}, \quad \tilde{\mathbf{n}} = \begin{bmatrix} \mathrm{Re}(\mathbf{n}) \\ \mathrm{Im}(\mathbf{n}) \end{bmatrix}.$$

This yields the real-valued system model

$$\tilde{\mathbf{y}} = \tilde{\mathbf{H}}\tilde{\mathbf{x}} + \tilde{\mathbf{n}}, \tag{2}$$

where $\tilde{\mathbf{y}} \in \mathbb{R}^{2N_r}$, $\tilde{\mathbf{x}} \in \mathbb{R}^{2N_t}$, $\tilde{\mathbf{H}} \in \mathbb{R}^{2N_r \times 2N_t}$, and $\tilde{\mathbf{n}} \in \mathbb{R}^{2N_r}$.

For notational simplicity, we omit the tildes in subsequent sections and refer to the real-valued system as

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

where $\mathbf{y} \in \mathbb{R}^M$, $\mathbf{x} \in \mathbb{R}^N$, $\mathbf{H} \in \mathbb{R}^{M \times N}$, and $\mathbf{n} \in \mathbb{R}^M$, with $M = 2N_r$ and $N = 2N_t$.

2.2. Problem Formulation

In MIMO detection, the objective is to recover the transmitted signal $\mathbf{x}$ given the received signal $\mathbf{y}$ and the known channel matrix $\mathbf{H}$. The optimal maximum likelihood (ML) detector is formulated as

$$\hat{\mathbf{x}}_{\text{ML}} = \arg\min_{\mathbf{x} \in \mathcal{X}^N} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2, \tag{4}$$

where $\mathcal{X}$ is the modulation alphabet. However, this combinatorial optimization problem is computationally prohibitive due to its exponential complexity with respect to N.

To address this issue, we adopt a regularized least squares approach suitable for deep unfolding. We formulate the detection problem as

$$\min_{\mathbf{x} \in \mathbb{R}^N} \frac{1}{2}\|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 + \lambda\|\mathbf{x}\|_1, \tag{5}$$

where $\lambda > 0$ is a regularization parameter. This optimization problem can be efficiently solved using the PGD algorithm, which proceeds iteratively as follows:

$$\mathbf{r}^{(k)} = \mathbf{x}^{(k-1)} - \eta_k \mathbf{H}^T(\mathbf{H}\mathbf{x}^{(k-1)} - \mathbf{y}), \tag{6}$$

$$\mathbf{x}^{(k)} = \text{prox}_{\lambda_k}(\mathbf{r}^{(k)}), \tag{7}$$

where $\mathbf{r}^{(k)}$ is the intermediate residual after the gradient update, η_k is the step size at the k-th iteration, $\mathbf{x}^{(k)}$ is the estimated transmitted signal at layer k, and $\text{prox}_{\lambda_k}(\cdot)$ is the soft-thresholding operator, defined component-wise as

$$\left[\text{prox}_{\lambda_k}(\mathbf{r}^{(k)})\right]_n = \begin{cases} r_n^{(k)} - \lambda_k & \text{if } r_n^{(k)} > \lambda_k, \\ 0 & \text{if } |r_n^{(k)}| \leq \lambda_k, \\ r_n^{(k)} + \lambda_k & \text{if } r_n^{(k)} < -\lambda_k, \end{cases} \tag{8}$$

with λ_k serving as the threshold parameter and $r_n^{(k)}$ denotes the n-th element of $\mathbf{r}^{(k)}$

3. Proposed Method

The corresponding deep unfolding network of the PGD algorithm, termed PGD-Net, has demonstrated significant performance improvements in MIMO detection by incorporating learnable parameters to optimize iterative updates.

Algorithm 1 KDE - MMD Quantization Aware Training

Initialize deep unfolding network with FP32 precision
Collect activation statistics $\{\mathbf{x}^{(k)}\}$ over training set
for each layer $k = 1$ to K **do**
  Kernel form calculation via KDE
  Initialize network parameters $\Theta^{(k)}$ and quantization steps $\Delta^{(k)}$
end for
while not converged **do**
  Forward pass:
    Compute $\mathbf{x}_S^{(k)} = Q_b(\mathbf{x}^{(k)}; \Delta^{(k)})$
    Calculate $\mathcal{L}_{\text{total}}$ via (16)
  Backward pass:
    Compute gradients $\nabla_{\Theta^{(k)}} \mathcal{L}_{\text{total}}$ via STE
    Update $\Theta^{(k)}$ and $\Delta^{(k)}$ using Adam optimizer
  Adjust $\Delta^{(k)}$ based on current SNR
end while

3.1. PGD Deep Unfolding for MIMO Detection

The transformation of the iterative PGD algorithm into a deep unfolded network (PGD-Net) involves systematically unrolling its update steps into a layered neural architecture, where each iteration corresponds to a dedicated network layer. Specifically, the gradient descent step

$$\mathbf{r}^{(k)} = \mathbf{x}^{(k-1)} - \eta_k \mathbf{H}^T(\mathbf{H}\mathbf{x}^{(k-1)} - \mathbf{y})$$

is implemented as a linear transformation layer with learnable step sizes $\{\eta_k\}$, while the proximal operator $\text{prox}_{\lambda_k}(\cdot)$ interpreted as a sparsity-inducing activation function—is embedded as a nonlinear module with trainable thresholds $\{\lambda_k\}$. By parameterizing η_k and λ_k as layer-specific weights rather than fixed hyperparameters, PGD-Net enables end-to-end gradient-based optimization, allowing the network to adaptively refine both the gradient direction and sparsity constraints across iterations (Mou et al., 2022). Crucially, the intermediate residual $\mathbf{r}^{(k)}$ and signal estimate $\mathbf{x}^{(k)}$ are propagated sequentially through the layers, mimicking the iterative flow of PGD while integrating data-driven feature learning.

However, deploying PGD-Net to resource-constrained environments, such as base stations and edge devices, faces computational bottlenecks due to its inherent complexity, necessitating efficient quantization strategies. While quantization-aware training (QAT) is conventionally tailored for lightweight models like deep unfolding networks (Wu et al., 2023), designing activation quantization loss functions remains challenging: conventional approaches relying on KL divergence or task-specific MSE often inadequately capture quantization discrepancies due to their reliance on restrictive distributional assumptions.

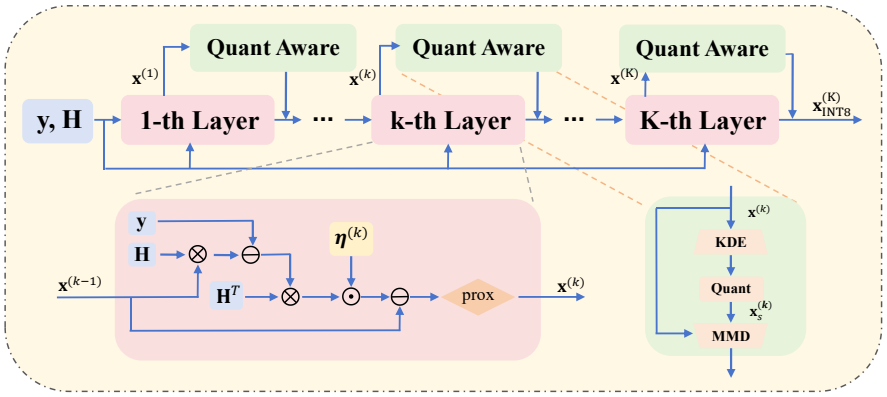

Figure 2. Architecture of the unfolded network, with details on the feed forward process of the k-th layer and the quantization aware module.

3.2. KDE-MMD Loss

To address this, we propose the KAQ framework. We first conducted a fit of the quantized activation values using a Gaussian kernel (Jayasumana et al., 2015; Kim et al., 2023). By setting $\sigma^{(k)}$ within the Gaussian kernel as learnable parameters, we designed a loss function in conjunction with the MMD method. This enabled the distribution of the quantized activation values to progressively approach their true distribution during training, thereby substantially enhancing the accuracy of quantization - aware training.

The quantization is performed as:

$$\mathbf{x}_S^{(k)} = Q_b(\mathbf{x}^{(k)}), \quad Q_b(\mathbf{x}^{(k)}) = \Delta^{(k)} \cdot \text{round}\left(\frac{\mathbf{x}^{(k)}}{\Delta^{(k)}}\right), \tag{9}$$

where b is the bit-width (8 in our case), and the initial step-size is:

$$\Delta^{(k)} = \frac{\max(|\mathbf{x}^{(k)}|)}{2^{b-1}-1}. \tag{10}$$

To enable gradient-sensitive optimization that adaptively minimizes the divergence between full-precision and quantized model, we define the maximum mean discrepancy (MMD) loss at layer k as:

$$\begin{aligned}\mathcal{L}_{\text{MMD}}^{(k)} &= \text{MMD}^2\left(\{\mathbf{x}_i^{(k)}\}_{i=1}^B, \{\mathbf{x}_{S,j}^{(k)}\}_{j=1}^B\right) \\ &= \frac{1}{B^2}\sum_{i,j}\mathcal{K}(\mathbf{x}_i^{(k)}, \mathbf{x}_j^{(k)}) + \frac{1}{B^2}\sum_{i,j}\mathcal{K}(\mathbf{x}_{S,i}^{(k)}, \mathbf{x}_{S,j}^{(k)}) \\ &\quad - \frac{2}{B^2}\sum_{i,j}\mathcal{K}(\mathbf{x}_i^{(k)}, \mathbf{x}_{S,j}^{(k)}),\end{aligned} \tag{11}$$

where B denotes the batch size. Taking the first term in the MMD loss as an example,

$$\frac{1}{B^2}\sum_{i,j}\mathcal{K}(\mathbf{x}_i^{(k)}, \mathbf{x}_j^{(k)}) = \frac{1}{B^2}\sum_{i,j}\exp\left(-\frac{\|\mathbf{x}_i^{(k)} - \mathbf{x}_j^{(k)}\|^2}{2(\sigma_1^{(k)})^2}\right), \tag{12}$$

where the Gaussian kernel $\mathcal{K}(\cdot,\cdot)$ is formulated by tow parts. From the distance metric perspective, $\|\mathbf{x}_i^{(k)} - \mathbf{x}_j^{(k)}\|^2$ computes the squared Euclidean distance between activation pairs from the full-precision model, measuring their pairwise dissimilarity. The adaptive bandwidth $\sigma_1^{(k)}$ serves as a data-driven scale parameter, for the second and third term of the MMD loss, the trainable parameters are demonstrated as $\sigma_2^{(k)}$ and $\sigma_3^{(k)}$. This method ensures that the model is able to learn the accurate activation distribution during training.

The gradients for quantized activations are computed as:

$$\begin{aligned}\frac{\partial \mathcal{L}_{\text{MMD}}}{\partial \mathbf{x}_{S,j}^{(k)}} &= \frac{1}{B^2}\sum_{i=1}^{B}\frac{\partial \mathcal{K}(\mathbf{x}_{S,j}^{(k)}, \mathbf{x}_{S,i}^{(k)})}{\partial \mathbf{x}_{S,j}^{(k)}} \\ &\quad - \frac{2}{B^2}\sum_{i=1}^{B}\frac{\partial \mathcal{K}(\mathbf{x}_{S,j}^{(k)}, \mathbf{x}_i^{(k)})}{\partial \mathbf{x}_{S,j}^{(k)}},\end{aligned} \tag{13}$$

taking the first term as an example, with the kernel gradient derivative:

$$\frac{\partial \mathcal{K}(\mathbf{x}_{S,j}^{(k)}, \mathbf{x}_{S,i}^{(k)})}{\partial \mathbf{x}_{S,j}^{(k)}} = -\frac{\mathbf{x}_{S,j}^{(k)} - \mathbf{x}_{S,i}^{(k)}}{(\sigma_2^{(k)})^2}\mathcal{K}(\mathbf{x}_{S,j}^{(k)}, \mathbf{x}_{S,i}^{(k)}). \tag{14}$$

This approach mitigates the accuracy loss using traditional QAT. Comparing to simply using MSE as the loss function during training, this kernel-based MMD approach bridges the gaps of activation distribution between the quantized model and full precisions. The specific structure of the quantization aware module and its integration with the unfolded network are illustrated in Fig. 2.

3.3. Joint Optimization of Quantization Step-Size

The time-varying nature of MIMO channels also requires adaptive quantization strategies. Static quantization steps fail to adapt different SNR conditions. Given the dynamic channel conditions in MIMO systems, we jointly optimize

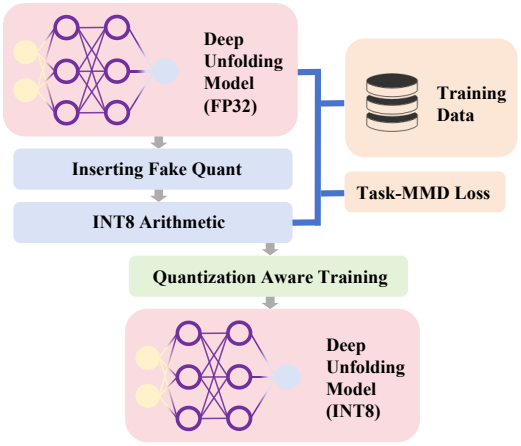

Figure 3. The KAQ training scheme from full precision model to quantized version

$\Delta^{(k)}$ based on SNR, formulated as:

$$\Delta^{(k)} = f_\theta(\text{SNR}) = \alpha \cdot \text{SNR}^{-1/2} + \gamma, \tag{15}$$

where the coefficients α, and γ are learnable parameters. This design captures that the $\text{SNR}^{-1/2}$ term reduces quantization step-sizes under high-noise conditions to preserve signal details. And the bias term γ maintains minimum quantization resolution even under ideal channel conditions.

3.4. Training with Quantization-aware Loss

The total loss combines the mean squared error (MSE) and the MMD loss, optimizing the detection accuracy and capturing the matching distribution of the activations at the same time,

$$\mathcal{L}_{\text{total}} = \underbrace{\frac{1}{B}\sum_{i=1}^{B} \|\mathbf{x}_{\text{true}} - \mathbf{x}_S^{(K)}\|_2^2}_{\text{Detection accuracy}} + \underbrace{\epsilon \sum_{k=1}^{K} \mathcal{L}_{\text{MMD}}^{(k)}}_{\text{Distribution matching}}, \tag{16}$$

where B is the batch size, and ϵ balances the terms. Network parameters $\Theta^{(k)}$ and $\Delta^{(k)}$ are updated via gradient descent:

$$\Theta^{(k)} \leftarrow \Theta^{(k)} - \mu \nabla_{\Theta_S} \mathcal{L}_{\text{total}}, \tag{17}$$

$$\Delta^{(k)} \leftarrow \Delta^{(k)} - \mu \nabla_{\Delta^{(k)}} \mathcal{L}_{\text{total}}, \tag{18}$$

where μ is the learning rate, enabling end-to-end quantization-aware training. For gradient propagation during training, we use the straight-through estimator (STE):

$$\frac{\partial Q_b(x)}{\partial x} = \begin{cases} 1 & \text{if } x \in [-\Delta^{(k)}(2^{b-1}-1), \Delta^{(k)}(2^{b-1}-1)], \\ 0 & \text{otherwise}, \end{cases} \tag{19}$$

which preserves gradient magnitude within the active quantization range while enabling quantization-aware training. Algorithm 1 summarizes the KAQ training approach while Fig. 3 illustrates the whole training scheme.

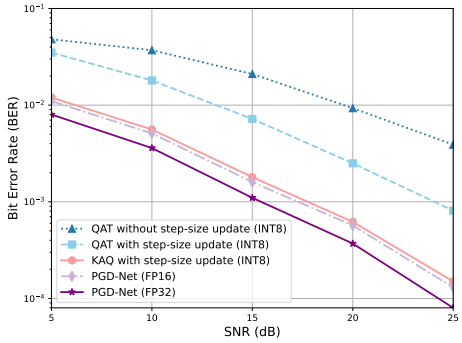

Figure 4. BER versus SNR.

4. Simulation Results

To assess the efficacy of the proposed KAQ framework for MIMO detection, numerical experiments are conducted at diverse angles. We compare the performance of quantized PGD-Net and ADMM-Net with and without the proposed KAQ framework. Furthermore, we compare the inference latency and memory storage of the quantized deep unfolding models with the full precision models.

The simulation setup employs a 16×16 MIMO system ($N_t = 16$ transmit and $N_r = 16$ receive antennas), corresponding to a real-valued system of dimensions $N = 32$ and $M = 32$. Transmitted symbols are generated from a 16-QAM constellation with independent real and imaginary components drawn from $\mathcal{X} = \{\pm 1, \pm 3\}$. The Rayleigh fading channel matrix $\mathbf{H}$ is configured with entries independently sampled from $\mathcal{N}(0, 1/N_r)$, while additive noise $\mathbf{n}$ is introduced at SNR levels ranging from 0 dB to 25 dB in 5-dB increments. This setup enables comprehensive evaluation under varying channel realizations and noise conditions.

The training protocol utilizes a dataset of 5×10^4 samples containing channel matrices $\mathbf{H}$, received signals $\mathbf{y}$, and transmitted symbols $\mathbf{x}$. Both PGD-Net and ADMM-Net architectures are unfolded into $K = 5$ trainable layers, initialized with uniform parameters $\eta_k = 0.1$ and $\lambda_k = 0.05$ for all layers k. Training is performed using the Adam optimizer with an initial learning rate of 10^{-3}, dynamically reduced by half every 10 epochs over a total of 50 training epochs. Mini-batches of size 128 are employed to balance computational efficiency and gradient estimation accuracy.

4.1. Model Accuracy

From the perspective of model accuracy, we first compare the bit error rate (BER) versus SNR of PGD-Net under different precisions. As shown in Fig. 4, the experimental results demonstrate that dynamically adjusting the quantization step size based on instantaneous SNR enables the quan-

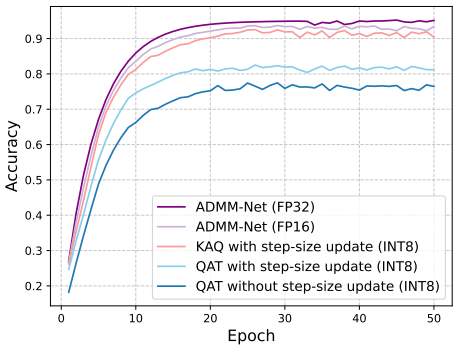

Figure 5. Model accuracy versus training epoch

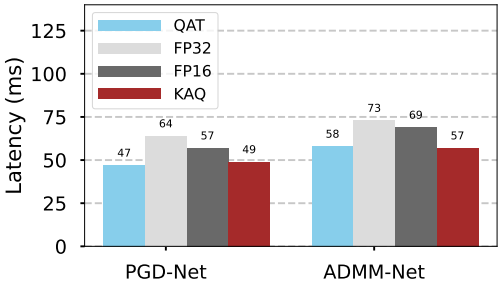

Figure 6. Comapision of inference latency

tized PGD-Net to achieve lower BER. When implementing the MMD loss, the incorporation of Gaussian kernels and learnable parameters $\sigma^{(k)}$ provides distinct advantages over conventional QAT approaches that solely employ MSE loss. Specifically, the MMD loss effectively captures distributional discrepancies induced by model quantization through its kernel-based metric. During the parameter update process of $\sigma^{(k)}$, the activation distributions progressively approximate the true distribution starting from an initial Gaussian assumption. This adaptive mechanism allows the KAQ dynamic quantization method to perform nearly as well as the FP16-precision PGD-Net baseline. At the same time, it significantly surpasses the traditional QAT method that depend on MSE loss optimization. In Fig. 5, it also can be observed that implementing the KAQ training scheme to quantize the ADMM-Net, can converge to a higher accuracy level compared to the traditional QAT method.

4.2. Inference Latency

Next, we illustrate the inference latency of both PGD-Net and ADMM-Net under different precisions. As shown in Fig. 6, the inference latency of the quantized models using KAQ is reduced in 20% compared to the full precision version. While keeping at the same level of inference latency with traditional QAT method, KAQ significantly outperforms on the model accuracy according to Section 4.1.

5. Conclusions

In this paper, we proposed a kernel-based adaptive quantization approach for model-driven deep unfolding networks. Specifically, we used a learnable Gaussian kernel that cooperated with the MMD approach to align the quantization-aware training loss. By further intrduced the dynamic quantization step-size, the proposed KAQ framework can be adaptive to wireless communication tasks based on the SNR conditions. Simulation results showed that the kernel-based adaptive quantization method achieves better accuracy performance than traditional QAT methods and efficiently reduces the inference time compares to the full precision models.

Impact Statement

Recent interest in efficient model deployment and edge computing has grown significantly, especially for deep unfolding models, driven by the ability of mathematical guidance. This work focuses on improving the accuracy and reducing the inference latency while implementing quantization through a novel kernel-based perspective, which has the potential to positively impact communication society. Furthermore, it connects model-driven deep learning with kernel-based learning, also has the potential on expanding its application in various types of deep neural networks.

References

Bai, T., Pan, C., Deng, Y., Elkashlan, M., Nallanathan, A., and Hanzo, L. Latency minimization for intelligent reflecting surface aided mobile edge computing. *IEEE Journal on Selected Areas in Communications*, 2020.

Barbarossa, S., Comminiello, D., Grassucci, E., Pezone, F., Sardellitti, S., and Di Lorenzo, P. Semantic communications based on adaptive generative models and information bottleneck. *IEEE Communications Magazine*, 2023.

Chang, T.-H., Hong, M., Liao, W.-C., and Wang, X. Asynchronous distributed admm for large-scale optimization—part i: Algorithm and convergence analysis. *IEEE Transactions on Signal Processing*, 2016.

Chen, M., Shao, W., Xu, P., Wang, J., Gao, P., Zhang, K., Qiao, Y., and Luo, P. Efficientqat: Efficient quantization-aware training for large language models. *arXiv preprint arXiv:2407.11062*, 2024.

Elbamby, M. S., Perfecto, C., Liu, C.-F., Park, J., Samarakoon, S., Chen, X., and Bennis, M. Wireless edge

computing with latency and reliability guarantees. *Proceedings of the IEEE*, 2019.

Elgabli, A., Park, J., Bedi, A. S., Bennis, M., and Aggarwal, V. Gadmm: Fast and communication efficient framework for distributed machine learning. *Journal of Machine Learning Research*, 2020.

Gong, R., Liu, X., Jiang, S., Li, T., Hu, P., Lin, J., Yu, F., and Yan, J. Differentiable soft quantization: Bridging full-precision and low-bit neural networks. In *ICCV*, 2019.

Han, R., Wang, S., Wang, S., Zhang, Z., Chen, J., Lin, S., Li, C., Xu, C., Eldar, Y. C., Hao, Q., and Pan, J. Neupan: Direct point robot navigation with end-to-end model-based learning. *IEEE Transactions on Robotics*, 2025.

Han, S., Mao, H., and Dally, W. J. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. In *ICLR*, 2016.

Hu, F., Deng, Y., Saad, W., Bennis, M., and Aghvami, A. H. Cellular-connected wireless virtual reality: Requirements, challenges, and solutions. *IEEE Communications Magazine*, 2020.

Hu, Q., Gao, F., Zhang, H., Li, G. Y., and Xu, Z. Understanding deep mimo detection. *IEEE Transactions on Wireless Communications*, 2023.

Hua, E., Jiang, C., Lv, X., Zhang, K., Ding, N., Sun, Y., Qi, B., Fan, Y., Zhu, X., and Zhou, B. Fourier position embedding: Enhancing attention's periodic extension for length generalization. In *ICML*, 2025.

Huang, W., Liu, Y., Qin, H., Li, Y., Zhang, S., Liu, X., Magno, M., and Qi, X. Billm: Pushing the limit of post-training quantization for llms. In *ICML*, 2024.

Huang, W., Qin, H., Liu, Y., Li, Y., Liu, X., Benini, L., Magno, M., and Qi, X. Slim-llm: Salience-driven mixed-precision quantization for large language models. In *ICML*, 2025.

Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., Adam, H., and Kalenichenko, D. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *CVPR*, 2018.

Jayasumana, S., Hartley, R., Salzmann, M., Li, H., and Harandi, M. Kernel methods on riemannian manifolds with gaussian rbf kernels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2015.

Jeong, E., Oh, S., Kim, H., Park, J., Bennis, M., and Kim, S.-L. Communication-efficient on-device machine learning: Federated distillation and augmentation under non-iid private data. In *NeurIPS Workshop on Machine Learning on the Phone and other Consumer Devices (MLPCD)*, 2018.

Johnston, J., Li, Y., Lops, M., and Wang, X. Admm-net for communication interference removal in stepped-frequency radar. *IEEE Transactions on Signal Processing*, 2021.

Khobahi, S., Shlezinger, N., Soltanalian, M., and Eldar, Y. C. Lord-net: Unfolded deep detection network with low-resolution receivers. *IEEE Transactions on Signal Processing*, 2021.

Kim, J., Lee, Y., and Liang, Z. The geometry of nonlinear embeddings in kernel discriminant analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

Lei, J., Zhou, J., Jian, Z., Liu, H., Zhang, L., Feng, Y., and Wu, Z. A wiener filter denoising based intelligent modulation recognition system. In *ICCC*, 2022.

Lei, J., Li, Y., Yung, L.-Y., Leng, Y., Lin, Q., and Wu, Y.-C. Understanding complex-valued transformer for modulation recognition. *IEEE Wireless Communications Letters*, 2024.

Li, D., Yang, K., and Wong, W. H. Density estimation via discrepancy based adaptive sequential partition. In *NeurIPS*, 2016.

Li, Z., Shi, W., and Yan, M. A decentralized proximal-gradient method with network independent step-sizes and separated convergence rates. *IEEE Transactions on Signal Processing*, 2019.

Lin, Q., Li, Y., Kou, W.-B., Chang, T.-H., and Wu, Y.-C. Communication-efficient activity detection for cell-free massive mimo: An augmented model-driven end-to-end learning framework. *IEEE Transactions on Wireless Communications*, 2024a.

Lin, Q., Li, Y., Wu, Y.-C., and Zhang, R. Intelligent reflecting surface aided activity detection for massive access: Performance analysis and learning approach. *IEEE Transactions on Wireless Communications*, 2024b.

Mou, C., Wang, Q., and Zhang, J. Deep generalized unfolding networks for image restoration. In *CVPR*, 2022.

Park, J., Samarakoon, S., Bennis, M., and Debbah, M. Wireless network intelligence at the edge. *Proceedings of the IEEE*, 2019.

Popovski, P., Nielsen, J. J., Stefanovic, C., Carvalho, E. d., Strom, E., Trillingsgaard, K. F., Bana, A.-S., Kim, D. M., Kotaba, R., Park, J., and Sorensen, R. B. Wireless access

for ultra-reliable low-latency communication: Principles and building blocks. *IEEE Network*, 2018.

Ren, Z., Lin, Q., Lei, J., Li, Y., and Wu, Y.-C. Mixture of experts-augmented deep unfolding for activity detection in irs-aided systems. *arXiv*, 2025.

Scardapane, S., Comminiello, D., Scarpiniti, M., and Uncini, A. Online sequential extreme learning machine with kernels. *IEEE Transactions on Neural Networks and Learning Systems*, 2015.

Shang, C., Tang, Y., Huang, J., Bi, J., He, X., and Zhou, B. End-to-end structure-aware convolutional networks for knowledge base completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019.

Shao, M. and Ma, W.-K. Binary mimo detection via homotopy optimization and its deep adaptation. *IEEE Transactions on Signal Processing*, 2021.

Tolstikhin, I., Sriperumbudur, B. K., and Schölkopf, B. Minimax estimation of maximum mean discrepancy with radial kernels. In *NeurIPS*, 2016.

Wang, Z., Pan, C., Huang, Y., Jin, S., and Caire, G. Randomized iterative algorithms for distributed massive mimo detection. *IEEE Transactions on Signal Processing*, 2025.

Wu, H., He, R., Tan, H., Qi, X., and Huang, K. Vertical layering of quantized neural networks for heterogeneous inference. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45, 2023.

Xiao, G., Lin, J., Seznec, M., Wu, H., Demouth, J., and Han, S. Smoothquant: Accurate and efficient post-training quantization for large language models. In *ICML*, 2023.

Xu, Y., Zhou, H., and Deng, Y. Task-oriented semantics-aware communication for wireless uav control and command transmission. *IEEE Communications Letters*, 2023.

Ye, Y., Chen, H., Xiao, M., Skoglund, M., and Vincent Poor, H. Privacy-preserving incremental admm for decentralized consensus optimization. *IEEE Transactions on Signal Processing*, 2020.

Yilmaz, S. F., Niu, X., Bai, B., Han, W., Deng, L., and Gündüz, D. High perceptual quality wireless image delivery with denoising diffusion models. In *INFOCOM WKSHPS*, 2024.

Yu, J., Zhou, R., Chen, C., Li, B., and Dong, F. Asfl: Adaptive semi-asynchronous federated learning for balancing model accuracy and total latency in mobile edge networks. In *ICPP*, 2023.

Zhang, C., Cao, X., Liu, W., Tsang, I., and Kwok, J. Nonparametric teaching for multiple learners. In *NeurIPS*, 2023a.

Zhang, C., Cao, X., Liu, W., Tsang, I., and Kwok, J. Nonparametric iterative machine teaching. In *ICML*, 2023b.

Zhang, C., Luo, S., Li, J., Wu, Y.-C., and Wong, N. Nonparametric teaching of implicit neural representations. In *ICML*, 2024.

Zhang, C., Bu, W., Ren, Z., Liu, Z., Wu, Y.-C., and Wong, N. Nonparametric teaching for graph property learners. In *ICML*, 2025.

Zhao, L., Wang, Y., Chu, X., Song, S., Deng, Y., Nallanathan, A., and Karagiannidis, G. K. Open-source edge ai for 6g wireless networks. *IEEE Network*, 2025.