MoE-CE: Enhancing Generalization for Deep Learning based Channel Estimation via a Mixture-of-Experts Framework

Tianyu Li [1] **Yan Xin** [1] **Charlie Zhang** [1]

Abstract

Reliable channel estimation (CE) is fundamental for robust communication in dynamic wireless environments, where models must generalize across varying conditions such as signal-to-noise ratios (SNRs), the number of resource blocks (RBs), and channel profiles. Traditional deep learning (DL)-based methods struggle to generalize effectively across such diverse settings, particularly under multitask and zero-shot scenarios. In this work, we propose MoE-CE, a flexible mixture-of-experts (MoE) framework designed to enhance the generalization capability of DL-based CE methods. MoE-CE provides an appropriate inductive bias by leveraging multiple expert subnetworks, each specialized in distinct channel characteristics, and a learned router that dynamically selects the most relevant experts per input. This architecture enhances model capacity and adaptability without a proportional rise in computational cost while being agnostic to the choice of the backbone model and the learning algorithm. Through extensive experiments on synthetic datasets generated under diverse SNRs, RB numbers, and channel profiles, including multitask and zero-shot evaluations, we demonstrate that MoE-CE consistently outperforms conventional DL approaches, achieving significant performance gains while maintaining efficiency.

1. Introduction

In modern wireless communication systems, reliable data transmission over time-varying and multi-path fading channels depends critically on the accurate knowledge of the channel state information (CSI). Channel estimation (CE) plays a vital role in this context by enabling the receiver to mitigate distortions introduced by the wireless medium. CE techniques are particularly important in advanced communication systems such as the fifth generation (5G) cellular networks, massive multiple input multiple output (MIMO), and millimeter-wave communications, where maintaining high data rates, low latency, and spectral efficiency is essential despite complex propagation conditions. Traditional channel estimation methods, including least squares (LS) and minimum mean square error (MMSE) (Neumann et al., 2018), rely heavily on pilot symbols and statistics of the channel models. However, the growing complexity of modern wireless environments has motivated the exploration of data-driven and learning-based approaches that can capture intricate channel behaviors and adapt to dynamic conditions more effectively. Nevertheless, these models often struggle to generalize across diverse deployment scenarios, such as varying signal-to-noise ratio (SNR) levels, the number of resource blocks (RBs), or channel profiles, particularly when task distribution shifts significantly at inference time.

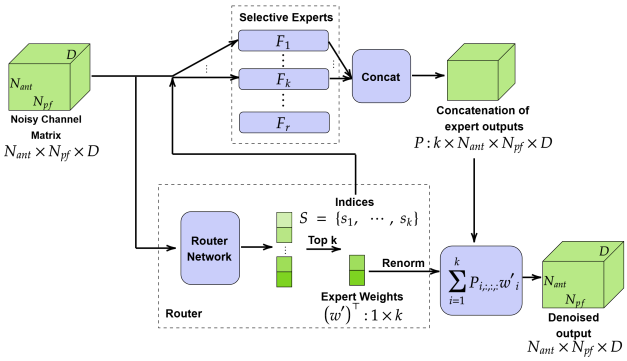

Figure 1. MoE framework for channel estimation.

Several efforts have been made to enhance the generalization capabilities of deep learning (DL)-based CE approaches. From a data-centric standpoint, (Luan & Thompson, 2023) highlights the critical role of training dataset design in achieving robust performance across varying channel conditions. Their work shows that exposing DL models to a rich diversity of simulated channel environments during training

[1]Standards and Mobility Innovation Lab, Samsung Research America, Berkeley Heights, New Jersey, USA. Correspondence to: Tianyu Li <tianyu.li@partner.samsung.com>.

Proceedings of the 42nd International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025. Copyright 2025 by the author(s).

significantly improves their adaptability to unseen deployment scenarios. From a learning algorithm perspective, in (Mao et al., 2019), the authors introduce a meta-learning-based approach designed to improve channel estimation in orthogonal frequency division multiplexing (OFDM) systems called RoemNet. It employs a meta-learner that can adapt to varying channel conditions, demonstrating superior performance compared to traditional methods under diverse scenarios. Despite these promising directions, many existing approaches emphasize data and training strategies while neglecting the role of model architecture itself. Architectural inductive biases, such as modularity, or conditional computation, are rarely explored as a primary avenue for improving generalization. Designing models that inherently support robustness to channel variability, even under standard training regimes, remains an open and underexplored challenge in the literature.

Mixture-of-experts (MoE) is a neural network architecture that introduces conditional computation by dynamically selecting a subset of specialized sub-models, or "experts", for each input. Originally proposed in (Jacobs et al., 1991), the core idea is rooted in the divide-and-conquer principle, where a complex problem is decomposed into smaller, more manageable subproblems. Unlike monolithic models that must use a shared set of weights for all data variations, MoE architectures encourage modularity, allowing different experts to specialize in distinct subspaces of the task distribution. This architectural inductive bias not only improves efficiency but also enhances the model's ability to generalize across heterogeneous tasks and input domains. In multitask setting, MoE has demonstrated strong performance by capturing task-specific structure while maintaining flexibility, as shown in domains such as robotics robotics (Huang et al., 2025) and computer vision (Chen et al., 2023). This architectural flexibility makes MoE particularly well-suited for zero-shot generalization as well, where the model must handle tasks or channel conditions that were not explicitly seen during training. The routing mechanism enables input-dependent expert selection, allowing the model to adaptively decompose complex learning problems and respond flexibly to new or unseen scenarios, as demonstrated in (Muqeeth et al., 2024). This property is especially valuable in domains like wireless channel estimation, where rapid adaptation to changing environments is critical and explicit supervision for every possible condition is impractical.

In this paper, we propose mixture-of-experts framework for channel estimation (MoE-CE), an MoE framework designed to enhance the generalization capability of DL-based channel estimation methods. The MoE-CE architecture comprises multiple expert sub-networks, each specializing in different channel characteristics or task variations, alongside a gating mechanism that dynamically routes input features to the most relevant experts. This modular design not only expands model capacity without a proportional increase in computational complexity but also enables flexible adaptation to a wide range of channel conditions.

Importantly, MoE-CE is agnostic to the specific learning algorithm and backbone network. In other words, it can incorporate any deep learning architecture for channel estimation and be trained using a variety of optimization strategies, from standard gradient-based methods with different optimizers to more advanced schemes like meta-learning (such as MAML (Finn et al., 2017)), to further improve the model's generalization capabilities.

We demonstrate that MoE-CE achieves substantial performance gains in both multitask learning settings, where the model learns across multiple channel types or system configurations; and in zero-shot scenarios, where it generalizes to previously unseen channel conditions at test time. Our results show that the combination of expert specialization and dynamic routing in MoE-CE is particularly well-suited for the non-stationary and diverse nature of wireless channels.

Contributions. The key contributions of this work are outlined as follows:

- We introduce MoE-CE, a flexible and learning-algorithm-agnostic framework that enhances the generalization performance of DL-based channel estimation. It supports arbitrary backbone architectures and can be integrated with various learning strategies, including meta-learning, to further boost generalization.
- We evaluate MoE-CE under both multitask and zero-shot settings, showing consistent improvements over conventional DL-based channel estimation (DL-CE) baselines under similar computational complexity.
- Using a mixed-SNR training scenario as a case study, we analyze expert selection patterns and demonstrate how MoE-CE promotes specialization and adaptability across diverse channel conditions.

2. Background and Related Work

To contextualize our proposed framework, this section reviews the foundational concepts and existing literature relevant to our work. We begin by presenting the mathematical formulation of the channel estimation problem in OFDM systems and briefly review recent DL-based approaches developed for this task. Next we introduce the MoE architecture, highlighting its core principles, practical applications in large-scale models, and relevance to channel estimation. Finally, we discuss strategies for managing expert load balancing, including both auxiliary-loss-based and auxiliary-loss-free approaches, which are critical for efficient and stable training of MoE-based systems.

2.1. Channel Estimation

OFDM is a widely used modulation technique in modern communication systems due to its robustness against frequency-selective fading and efficient implementation via the Fast Fourier Transform (FFT). In this section, we describe the mathematical formulation of the channel estimation problem in a typical OFDM system.

Consider an OFDM system with N_{pf} pilot sub-carriers and N_{ant} receive antennas. The input-output relationship between transmitted and received signals at pilot sub-carriers in the frequency domain can be written as:

$$\mathbf{Y} = \mathbf{H} \odot \mathbf{X} + \mathbf{W}, \tag{1}$$

where $\mathbf{Y} \in \mathbb{C}^{N_{ant} \times N_{pf}}$ are the received signals at N_{pf} pilot sub-carriers and N_{ant} receive antennas, $\mathbf{H} \in \mathbb{C}^{N_{ant} \times N_{pf}}$ denotes the channel matrix in the space-frequency domain, the operator $\odot$ represents the Hadamard product that is an element-wise product, $\mathbf{X} \in \mathbb{C}^{N_{ant} \times N_{pf}}$ are the transmitted pilot signals known to the receiver, and $\mathbf{W} \in \mathbb{C}^{N_{ant} \times N_{pf}}$ is an additive white Gaussian noise (AWGN).

The goal of the channel estimation task is to estimate channel matrix $\mathbf{H}$ based on the pilot signal $\mathbf{X}$ and the received signals $\mathbf{Y}$. The simplest channel estimation solution is the LS estimate, denoted by $\hat{\mathbf{H}}^{LS}$, which is readily computed by:

$$\hat{\mathbf{H}}^{LS}_{i,j} = \frac{\mathbf{Y}_{i,j}}{\mathbf{X}_{i,j}}, \ \forall i \in [N_{ant}], \ j \in [N_{pf}], \tag{2}$$

where the notation $[N]$ denotes all positive integers no larger than N. The above equation can therefore be further simplified to:

$$\hat{\mathbf{H}}^{LS} = \mathbf{H} + \mathbf{W}. \tag{3}$$

2.2. DL-based Channel Estimation

In recent years, DL has emerged as a powerful tool in the field of wireless communications, offering promising enhancements to traditional signal processing algorithms. Various DL-based techniques have been explored to improve tasks such as modulation classification (O'Shea & Hoydis, 2017), signal detection (Erdogmus et al., 2001), channel equalization (He et al., 2018), and CSI feedback compression (Samuel et al., 2017). In the context of channel estimation, conventional estimators like LS and MMSE rely on statistical assumptions and predefined models, which may not generalize well to real-world environments. In contrast, DL-based methods can learn complex mappings directly from data, enabling more flexible and robust estimation. DL-based channel estimation techniques can be broadly divided into two categories. The first category adopts an end-to-end learning perspective, treating the entire communication system as a differentiable model. For instance, in (Ye et al., 2017), a deep neural network is trained to perform encoding, decoding, channel estimation and all other functionalities of a communication link jointly in an implicit fashion, showing significant performance gains over traditional baselines. Similarly, (O'Shea & Hoydis, 2017) proposes an autoencoder-based communication system that integrates modulation, channel estimation, and decoding into a single trainable model. However, such approaches often lack explicit access to the estimated channel state, thereby limiting their applicability in systems where CSI is needed for other signal processing tasks. The second category focuses specifically on learning the channel matrix using supervised deep neural networks. In (Wen et al., 2018), the authors treat the channel matrix as a two-dimensional (2D) image and apply a convolutional neural network (CNN) (LeCun et al., 1998) for denoising-based channel estimation in massive MIMO systems. This method captures spatial correlations across antennas effectively. Following this idea, many recent channel estimation works have adopted this framework, such as (Li et al., 2020; Soltani et al., 2019; Ahmad et al., 2023; Hu et al., 2021; Dong et al., 2019), proving the importance and effective of this DL framework for channel estimation.

From the Equation (3), the CE problem can be viewed as a 2D denoising problem in the frequency domain, where the goal is to recover the true channel matrix from its noisy LS estimate. Typically, a neural network is trained to learn a nonlinear mapping from the noisy LS estimate, obtained using pilot symbols, to the underlying clean channel matrix.

$$\hat{\mathbf{H}} = f_\theta(\hat{\mathbf{H}}^{LS}), \tag{4}$$

where f_θ is a neural network parameterized by θ.

By decomposing the complex channel matrix into its real and imaginary components, we obtain a tensor of size $N_{ant} \times N_{pf} \times 2$, effectively transforming the channel estimation task into an image denoising problem. Neural networks developed for vision tasks, such as CNNs (LeCun et al., 1998) and more advanced architectures like Resnet (He et al., 2016), and NAFNet (Chen et al., 2022), are well-suited for this setting, as they can effectively capture spatial correlations across antenna and subcarrier dimensions. When applied to channel estimation, these models are typically trained using the normalized mean squared error (NMSE) as the loss function:

$$\mathcal{L}_{\text{NMSE}} = \mathbb{E}\left[\frac{\|\mathbf{H} - \hat{\mathbf{H}}\|_2^2}{\|\mathbf{H}\|_2^2}\right]. \tag{5}$$

2.3. Mixture of Experts Architecture

Modern MoE architectures are often based on transformers and consist of two main elements: sparse MoE layers and a gating network or the so-called router. On one hand, the sparse MoE layer consists of various "expert" blocks.

Typically, these expert blocks are parameterized by feed forward networks (FFN) in the transformer architecture, often with the same structure. The router, on the other hand, determines which input is sent to which expert. The router does so by outputting a vector of weights for all the experts. A subset of the top-k experts is then selected based on these weights, and only these experts are activated to process the input. Finally, the outputs of the selected experts are aggregated and passed to the next layer. Notably, the MoE layer often replaces the traditional FFN layer in the Transformer architectures, enabling more specialized modeling without incurring too much computational overhead. Through expert specialization, the FFN in the MoE layer likely requires fewer parameters, thereby further enhancing the computational efficiency of the transformer.

In large-scale language models, MoE has been widely adopted to enhance scalability while maintaining computational efficiency. Models such as Switch Transformer (Fedus et al., 2022), GShard (Lepikhin et al.), and DeepSeek (Liu et al., 2024) utilize MoE architectures to increase the number of parameters without a proportional rise in computational cost. In these architectures, only a small fraction of the experts are active per token, significantly reducing the per-step float operations (FLOPs) consumption compared to a dense model of similar size. The key benefit of MoE in language models is its ability to scale efficiently while mitigating inference costs. By activating only a subset of experts, MoE architecture enables models to learn diverse representations across different tokens, capturing nuanced patterns in natural language.

An MoE layer with r experts is defined as follows: given an input $\boldsymbol{x}$, a set of expert functions $\{F_1, F_2, \ldots, F_r\}$ each computing a candidate output, and a gating function $R(\boldsymbol{x}) \in \mathbb{R}^r$ assigning routing weights, the output of a fully routed MoE layer is computed as:

$$\hat{\boldsymbol{y}} = \sum_{i=1}^{r} R(\boldsymbol{x})_i \cdot F_i(\boldsymbol{x}), \tag{6}$$

where $R(\boldsymbol{x})_i \geq 0$ and $\sum_{i=1}^{r} R(\boldsymbol{x})_i = 1$.

In the case of hard routing with top k selection, only the k experts with the highest gating scores are activated, reducing computational cost. The output becomes:

$$\hat{\boldsymbol{y}} = \sum_{i \in \mathcal{T}_k(\boldsymbol{x})} R(\boldsymbol{x})_i \cdot F_i(\boldsymbol{x}), \tag{7}$$

where $\mathcal{T}_k(\boldsymbol{x})$ denotes the indices of the top k experts selected by the gating function.

2.4. Load Balancing in MoE

Training MoE models requires careful handling of load balancing to prevent uneven expert utilization. When certain experts receive a disproportionate amount of traffic, the model's training efficiency and convergence may deteriorate. To address this, prior works have employed auxiliary losses to encourage balanced expert utilization. Notably, GShard (Lepikhin et al.) and Switch Transformer (Fedus et al., 2022) incorporate auxiliary losses that penalize imbalanced expert activation. For instance, GShard utilizes a load balancing loss that discourages excessive reliance on a small subset of experts. Additionally, Switch Transformer simplifies the gating mechanism to a top 1 selection, reducing the routing overhead while improving load distribution. Specifically, Switch Transformer utilizes the auxiliary loss:

$$\mathcal{L}_{\text{load}} = \sum_{i=1}^{r} \frac{\alpha \cdot N}{T^2} \sum_{\boldsymbol{x} \in \mathcal{B}} \mathbb{1}\{\text{argmax } R(\boldsymbol{x}) = i\} \sum_{\boldsymbol{x} \in \mathcal{B}} R(\boldsymbol{x})_i, \tag{8}$$

where $\mathcal{B}$ denote the current batch, T is the number of tokens, N is the number of examples and α is a regularization weight. Equation (8) encourages uniform routing since it is minimized under a uniform distribution.

2.5. Auxiliary-Loss-Free Load Balancing

To eliminate the need for explicitly tuning load balancing losses, recent studies have proposed architectural strategies that naturally promote balanced expert usage. One such approach, auxiliary loss-free load balancing (ALFLB), is introduced by DeepSeek (Guo et al., 2025; Liu et al., 2024; Wang et al., 2024). Instead of enforcing load balancing through auxiliary objectives in classic literature, this approach leverages an adaptive gating function that naturally distributes computation across experts. To achieve this, the model needs to maintain an expert bias, initialized as an all-zero vector of size r. After each gradient update, one needs to compute the frequency of the expert selection. If the frequency for an expert being selected is higher than a threshold τ_1, we call this expert being "over-utilized". Otherwise, if it is lower than a threshold τ_2, we refer to it as being "under-utilized". For the over-utilized experts, we decrease the corresponding expert bias by γ while for the under-utilized ones, we increase their expert bias by γ. When selecting the top k experts, we use the sum of the router's output weights and the expert bias to determine which experts to select.

3. Methodology

In this section, we present the mixture-of-experts framework for channel estimation (MoE-CE) in details. This framework is flexible and can accommodate any backbone ML models and learning methods. The core idea is to construct multiple expert networks, which can adopt various neural network architectures tailored to different aspects of CE, e.g. different SNR levels, RB numbers, or channel

Algorithm 1 Training MoE-CE with ALFLB and SGD

Experts (subnetworks): NN parameterized functions $F_1, \cdots, F_r : \mathbb{R}^{N_{ant} \times N_{pf} \times D} \to \mathbb{R}^{N_{ant} \times N_{pf} \times D}$.
Router: NN parameterized function $R : \mathbb{R}^{N_{ant} \times N_{pf} \times D} \to \mathbb{R}^r$.
Input: Noisy channel (LS estimate of the channle matrix) $\hat{\mathbf{H}}^{LS} \in \mathbb{R}^{N_{ant} \times N_{pf} \times D}$.
Input: Clean channel $\mathbf{H} \in \mathbb{R}^{N_{ant} \times N_{pf} \times D}$.
repeat
  Initialize the expert bias $\boldsymbol{u} \in \mathbb{R}_+^r$ for ALFLB to be all zeros.
  Initialize the router R and the subnetworks $F_1, \cdots, F_r$.
  Compute the forward pass of the router function: $\boldsymbol{w}^\top = R(\hat{\mathbf{H}}^{LS})$.
  Compute the top k selection weights $\tilde{\boldsymbol{w}}^\top = \boldsymbol{w}^\top + \boldsymbol{u}^\top$
  Obtain the top k indices based on $\tilde{\boldsymbol{w}}$, e.g., $S = \{1, \cdots, k\}$ and the corresponding router weights weights $\boldsymbol{w}'^\top$, e.g., $\boldsymbol{w}'^\top = [\boldsymbol{w}_1, \cdots, \boldsymbol{w}_k]^\top$.
  Based on the expert selection frequency, adjust the expert bias $\boldsymbol{u}$.
  Renormalize the weights: $\boldsymbol{w}'^\top = \frac{\boldsymbol{w}'^\top}{<\boldsymbol{w}'^\top, \mathbf{1}>}$.
  Compute the forward pass of the selected subnetworks, e.g., $F_1(\hat{\mathbf{H}}^{LS}), \cdots, F_k(\hat{\mathbf{H}}^{LS})$. Stack these candidate outputs to form a tensor $\mathbf{P} \in \mathbb{R}^{k \times N_{ant} \times N_{pf} \times D}$.
  Obtain the final output:

$$\sum_{i=1}^{k} \mathbf{P}_{i,:,:,:} \boldsymbol{w}'_i.$$

  Compute the NMSE loss function $\mathcal{L}_{\text{NMSE}}$ w.r.t. $\mathbf{H}$ and run stochastic gradient descent to update the parameters of the selected subnetworks $F_1, \cdots, F_k$ and the router R.
until Converge

profiles. A lightweight router network is responsible for dynamically selecting the top k most relevant experts during each forward pass, enabling efficient resource allocation and adaptive learning. By dynamically selecting the most suitable experts based on the current input, the model achieves improved generalization and computational efficiency, enhancing the robustness and adaptability of CE models in dynamic communication environments. We will show later in the experiment sections that this framework works well not only under a multitask set-up but also has significant performance gain when testing in a zero-shot setting as well.

Figure 1 illustrates the MoE-CE pipeline. An MoE-CE top k/r with r selective experts and top k selection goes as the following: first we take as input the LS estimate of the channel matrix either in the frequency domain or the delay domain, which is of size $N_{ant} \times N_{pf} \times D$. Note that typically D is set to 2 indicating the real and imaginary decomposition of the complex channel matrix. Alternatively, the parameter D can also be 4 when polarization is introduced. The input will first go through a router network $R : \mathbb{R}^{N_{ant} \times N_{pf} \times D} \to \mathbb{R}^r$, mapping the input data to a size r vector. Then after applying softmax function, we obtain the expert weights $\boldsymbol{w}^\top$ and subsequently select the top k experts based on the corresponding expert weights and obtain the selected expert indices $S = \{s_1, \cdots, s_k\}$ as well as their corresponding expert weights $\boldsymbol{w}'^\top \in \mathbb{R}^k$. For the ease of illustration, let us assume $s_i = i$, for all $i \in [k]$, i.e., $S = \{1, 2, \cdots, k\}$. Note that the expert selection varies based on the current input. After obtaining the selected expert indices, the LS estimate input will go through the forward pass of the selected expert networks, namely $F_1, \cdots, F_r : \mathbb{R}^{N_{ant} \times N_{pf} \times D} \to \mathbb{R}^{N_{ant} \times N_{pf} \times D}$, which can be parameterized as arbitrary neural networks. Assuming $S = \{1, 2, \cdots, k\}$, the input only goes through the selected k expert networks, i.e., $F_1, \cdots, F_k$ and for the remaining $r - k$ networks, no forward pass is carried out. We then obtain k candidate outputs of size $\mathbb{R}^{N_{ant} \times N_{pf} \times D}$ and concatenate them together to form a tensor $\mathbf{P} \in \mathbb{R}^{k \times N_{ant} \times N_{pf} \times D}$. The selected expert weights $\boldsymbol{w}'^\top$ are renormalized $\boldsymbol{w}'^\top = \frac{\boldsymbol{w}'^\top}{<\boldsymbol{w}'^\top, \mathbf{1}>}$, where $\mathbf{1}$ is an all 1 vector of size k, so that it sums up to 1. Finally, a weighted sum is computed to obtain the final denoised channel $\sum_{i=1}^{k} \mathbf{P}_{i,:,:,:} \boldsymbol{w}'_i$. For resolving load balancing issue, we opt to use ALFLB as introduced in the previous section. We further explain the complete training procedure of MoE-CE with ALFLB and stochastic gradient descent (SGD) in Algorithm 1. We defer the integration of MoE-CE with alternative learning schemes, such as meta-learning, for future work.

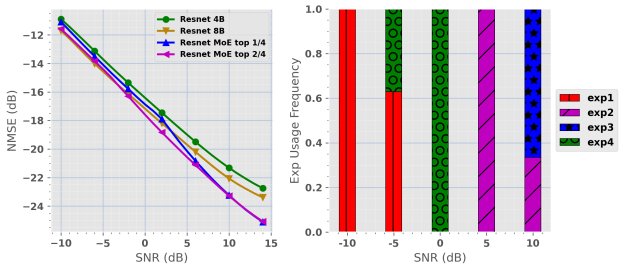

Figure 2. Cross SNR levels channel estimation performance comparison between vanilla Resnet and Resnet-MoE (left) and expert usage analysis (right).

4. Experiment

Generalizing to multiple SNR levels, channel profiles, and RB numbers is vital for building robust and scalable chan-

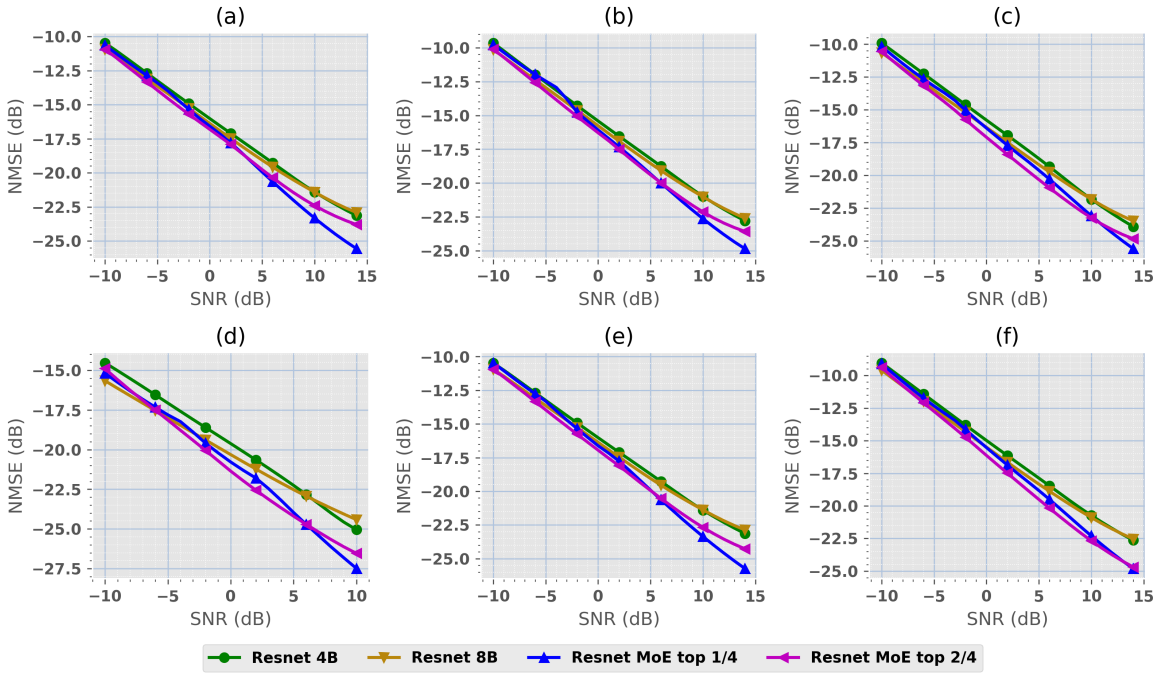

Figure 3. Cross channel profiles channel estimation performance comparison between vanilla Resnet and Resnet-MoE under multitask setting on: (a) UMi, (b) UMa, (c) CDL-B and (d) CDL-D; and under zero-shot generalization setting on: (e) CDL-B with 1200 ns delay spread and (f) UMi with 1200 delay spread.

nel estimation models in modern wireless communication systems. Real-world environments are dynamic, with fluctuating SNR and varying channel characteristics due to mobility, interference, and deployment scenarios. A model that can generalize across these variations ensures consistent performance without retraining, reducing latency, and improving system reliability. Likewise, generalization across RB numbers allows the model to adapt to different bandwidth allocations, enabling flexibility and efficiency across standards like 5G and future 6G systems. Such generalization not only lowers deployment complexity and cost by avoiding the need for specialized models but also supports long-term adaptability to evolving network conditions, spectrum usage, and hardware configurations.

When generalizing to different RB numbers or channel profiles, it is often necessary to also generalize to various SNR levels simultaneously because these dimensions of variability are inherently intertwined in real-world wireless environments. For example, a change in RB number affects frequency resolution and spectral efficiency, but its impact on performance is highly dependent on the SNR: what works well at high SNR may fail at low SNR due to significant noise level discrepancy. Similarly, different channel profiles (e.g., urban vs. rural or line-of-sight vs. non-line-of-sight) exhibit distinct multipath and fading characteristics, whose effects are amplified or diminished depending on the SNR. Therefore, to ensure robust channel estimation under realistic deployment scenarios, the model must be capable of handling the joint variability of RB number, channel profile, together with different SNR levels.

In this section, we conduct a series of experiments to evaluate the effectiveness and generalization capability of the proposed MoE-CE framework. We design our evaluations to reflect realistic challenges in wireless communication, including varying signal-to-noise ratios (SNRs), channel profiles, and RB numbers. For experiments involving diverse channel profiles and RB numbers, we also incorporate cross-SNR generalization to simulate more practical and demanding deployment scenarios. The experiments are organized to test both multitask learning performance and zero-shot generalization ability. We compare MoE-CE against strong deep learning baselines under matched computational budgets, and analyze expert utilization patterns to provide insights into the model's adaptability and specialization behavior across diverse conditions.

Throughout the experiment section, we use Adam (Kingma & Ba, 2014) with learning rate 0.001 as the optimizer. The training and test data are generated from physical uplink shared channel (PUSCH) using Siona (Hoydis et al., 2022). For ALFLB setup, we set $\tau_1 = \frac{2}{r}$, $\tau_2 = \frac{4}{5r}$ and $\gamma = 0.001$ via cross validation, where r is the number of selective

Table 1. Model computational complexity and size comparison between MoE-CE and vanilla DL methods. The complexity is computed based on an input size of $16 \times 240 \times 4$ (with polarization).

MODEL	MACS	FLOPS	#PARAMETERS	MODEL SIZE
RESNET MOE TOP $1/4$	79.43 M	158.86 M	81.19 K	317 KB
RESNET MOE TOP $2/4$	159.28 M	318.56 M	81.19 K	317 KB
RESNET 4B	78.84 M	157.68 M	19.99 K	78 KB
RESNET 8B	155.03 M	310.06 M	38.80 K	151 KB
NAFNET MOE TOP $1/4$	22.2 M	44.4 M	147.78 K	577 KB
NAFNET MOE TOP $2/4$	44.81 M	89.62 M	147.78 K	577 KB
NAFNET VANILLA	21.61 M	43.22 M	36.64 K	145 KB

experts. For all MoE-CE models presented in this section, we use a three-layer CNN as the router architecture, with 3×3 filter size and r, $2r$, and r hidden channels in the respective layers. The output of the CNN router is passed through global average pooling followed by a softmax function to produce the initial task weights.

4.1. Data Preprocessing and Postprocessing

The data we obtained from the Siona-based system level simulator are the clean channel matrix (the label) and the LS estimate (noisy) of the channel matrix, under the PUSCH frequency domain and OFDM format. We first convert the data into delay domain using fast Fourier transformation (FFT). The ML models then operate on the delay domain transformed data. After obtaining the output from the ML model, an inverse FFT (IFFT) is performed to recover the prediction back to the frequency domain. The loss function as well as the evaluation metric is then computed under the frequency domain.

4.2. Mixed SNRs

In this experiment, we showcase the generalization capability of the proposed MoE framework on cross SNRs setups. To be more specific, we generate synthetic training data from urban micro (UMi) channel profile, with SNR ranges from -10 dB to 12 dB, taken every 2 dB. The RB number for both training and test data is set to be 40. For evaluation, we look at the NMSE result on a separate test dataset of the same RB number and channel profile, with SNR ranges from -10 dB to 14 dB. For this experiment, we use Resnet (He et al., 2016) as the backbone model, namely all experts are parameterized by the Resnet architecture. Specifically, for baseline models, we use Resnet with 4 Resnet blocks (Resnet-4B) and 8 blocks (Resnet-8B), and the channel size is set to 16. For all MoE experts, we use Resnet-4B with 16 channel size. In the following experiments, without other specifications, all MoE experts are using Resnet-4B with 16 channel size as the architecture. We use 3×3 kernel size for all CNN layers in all experiments.

In the left figure of Figure 2, we show the NMSE comparison of our Resnet-MoE architecture with top 1 and top 2 expert selection out of 4 selective experts. The notation top $1/4$ means we are using top 1 selection, and the total number of selective experts is 4. Note that under this set-up, Resnet MoE top $1/4$ and $2/4$ share similar computation complexity to Resnet-4B and 8B, respectively. We can see clearly that MoE based models achieve better performance, especially under relatively high SNR cases.

Notice that the curve of top $1/4$ (blue curve with upward triangle markers) is not very smooth. This is because different experts are selected at different SNR levels. Due to the discrete nature of the top 1 selection, we can observe the sharp turning point on this curve. The right figure of Figure 2 illustrates the distribution of expert usage during evaluation across different SNR levels. At -10 dB SNR, the router chooses expert 1 for all evaluation input data. This behavior is gradually shifted to expert 4 at 0 dB SNR. Then at 5 dB SNR the router shifts the selection entirely to expert 2. This is subsequently changed to a mix of expert 2 and expert 3 at 10 dB SNR. Overall, experts 1 and 4 are often selected for low SNR cases while experts 2 and 3 are for high SNR ones.

In Table 1, we present the computational complexity of all models compared in the experiment section. One can clearly see that Resnet MoE top $1/4$ shares similar FLOPs with Resnet 4B, while Resnet MoE top $2/4$ shares similar FLOPs with Resnet 8B. One thing to note is that MoE-CE architecture leverages several subnetworks simultaneously, therefore it significantly increases the number of parameters and the model size.

4.3. Mixed SNRs and Channel Profiles

In this experiment, we want to evaluate the ability of the proposed MoE architecture to generalize to multiple SNR levels and channel profiles, such as UMi, urban macrocellular channel (UMa), clustered delay line (CDL)-B and CDL-D. To do this, we generate synthetic data from all these channel profiles under PUSCH, with SNR ranges from -10 dB to 12 dB. The RB number for both training and evaluation is still set to 40 and with delay spread set to 300 ns, 300 ns, 600 ns,

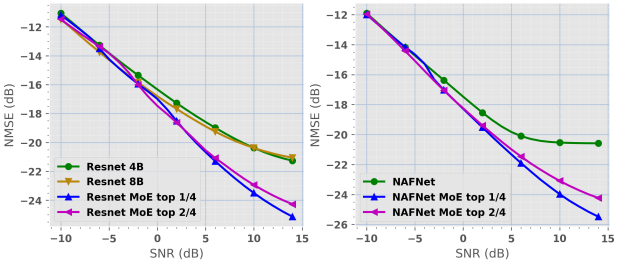

Figure 4. Zero-shot generalization performance comparison for varying RB numbers between vanilla Resnet and Resnet-MoE (left) and vanilla NAFNet and NAFNet-MoE (right).

10 ns for the respective channel profiles. For the expert's architecture, we follow the same Resnet structure as in the previous experiment (4 blocks with 16 channels). For a multitask setting, subfigures (a), (b), (c), and (d) of Figure 3 present evaluation result of the trained model on UMi, UMa, CDL-B, CDL-D with the same delay spread as the training data respectively. For a zero-shot setting, subfigures (e), (f) show the evaluation results of the same trained model but on CDL-B and UMi with 1200 ns delay spread instead. We can see that MoE-based approach consistently outperforms the vanilla method, under both multitask setting as well as zero-shot adaptation setting. The gain in performance is more significant with increased SNR, the same phenomenon as we observed in the previous experiment. An additional observation is that increasing the number of selected experts tend to enhance the performance of the MoE model in the low SNR regime, while causing a slight performance degradation in the higher regime. This highlights a trade-off that needs to be carefully considered when implementing an MoE-based architecture.

4.4. Mixed SNRs and Varying RB Numbers

In this experiment, besides showing the generalization capability of the MoE architecture to multiple SNR levels and RB numbers, we also showcase the flexibility of the proposed MoE structure. In addition to using Resnet as the backbone model, we also use a 15-layer NAFNet (Chen et al., 2022) with 8 hidden channels as the backbone model. For this experiment, we use the training data generated from a UMi channel with RB numbers of $5, 9, 12, 16, 20$ with SNR ranges from -10 dB to 12 dB. The validation data has the same setup as the training data. For evaluation, we generate data from the same UMi channel but with 54 RBs. For Resnet, compared between Resnet 4B, 8B and Resnet MoE top $1/4$, $2/4$. Same as before, Resnet MoE utilizes a Resnet 4B as the backbone model. Additionally, we also compare between vanilla NAFNet and NAFNet MoE of 4 selective experts with top 1 and top 2 selections. Note that this experiment is purely zero-shot setting as the test RB number does not exist in the training RB numbers.

Figure 4 presents the cross RB numbers performance evaluation of Resnet backbone (left) and NAFNet backbone (right) for testing the 54 RB configuration under the delay domain. We can see that in both cases there is a significant performance gain using the MoE-CE architecture compared to the vanilla model. In Table 1, we show the computational complexity and model size for NAFNet-MoE. We observe that, same as Resnet-MoE, the forward pass complexity between the vanilla NAFNet and NAFNet-MoE top $1/4$ is nearly identical, albeit the small router complexity. Though the model size and number of parameters is four times the vanilla NAFNet.

5. Conclusion

In this paper, we propose MoE-CE, a mixture-of-experts framework designed to enhance the generalization ability of DL-based channel estimation methods in wireless communication systems. By combining multiple expert networks with a lightweight router for dynamic expert selection, MoE-CE enables task-specific specialization while maintaining computational efficiency. We demonstrated that this architecture generalizes well across a wide range of scenarios, including varying SNR levels, RB numbers, and channel profiles, under both the multitask and zero-shot settings. Extensive experiments showed that MoE-CE consistently outperforms conventional DL-based methods, achieving improved accuracy with comparable computational cost. Furthermore, the framework's compatibility with diverse backbones and training strategies makes it a flexible and scalable solution for real-world deployment in dynamic and heterogeneous wireless environments.

Although the proposed MoE-CE framework exhibits robust generalization across varying SNR levels, RB numbers, and channel profiles, several promising avenues remain open for future research. One immediate possibility is to incorporate model-agnostic meta-learning (MAML) (Finn et al., 2017) or other meta learning algorithms into MoE-CE, further enhance zero-shot generalization capability of the framework. Furthermore, in the current setup, all experts share the same backbone structure. Designing heterogeneous or dynamically configurable expert architectures, e.g., various approaches mentioned in (Han et al., 2021), could allow for more fine-grained specialization, potentially improving both performance and efficiency. Last but not least, developing formal theoretical understanding of how and why experts specialize, especially under cross-condition generalization, could offer insights on the channel estimation problem under such system configuration, and additionally for more principled design of future MoE-based architectures for communication systems.

Impact Statement

This work aims to advance the field of machine learning for wireless communication by improving the generalization capability of deep learning-based channel estimation through a modular MoE framework. By enabling robust performance across diverse and dynamic environments, our approach has the potential to enhance the reliability and adaptability of future wireless systems, including 5G and beyond. Improved channel estimation may lead to more efficient use of spectrum and energy, benefiting both infrastructure providers, end users as well as the environment.

We do not anticipate any direct negative societal consequences or ethical concerns from this work. However, as with all technologies that improve the performance of communication systems, care must be taken to ensure equitable access and responsible deployment.

References

Ahmad, M., Shakeel, T., and Shin, S. Y. Image super resolution based channel estimation for future wireless communication. *Computer Networks*, 237:110057, 2023.

Chen, L., Chu, X., Zhang, X., and Sun, J. Simple baselines for image restoration. In *European Conference on Computer Vision*, pp. 17–33. Springer, 2022.

Chen, T., Chen, X., Du, X., Rashwan, A., Yang, F., Chen, H., Wang, Z., and Li, Y. AdaMV-MoE: Adaptive multi-task vision mixture-of-experts. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 17346–17357, 2023.

Dong, P., Zhang, H., Li, G. Y., Gaspar, I. S., and NaderiAlizadeh, N. Deep cnn-based channel estimation for mmwave massive MIMO systems. *IEEE Journal of Selected Topics in Signal Processing*, 13(5):989–1000, 2019. doi: 10.1109/JSTSP.2019.2925975.

Erdogmus, D., Rende, D., Principe, J. C., and Wong, T. F. Nonlinear channel equalization using multilayer perceptrons with information-theoretic criterion. In *Neural Networks for Signal Processing XI: Proceedings of the 2001 IEEE Signal Processing Society Workshop (IEEE Cat. No. 01TH8584)*, pp. 443–451. IEEE, 2001.

Fedus, W., Zoph, B., and Shazeer, N. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pp. 1126–1135. PMLR, 2017.

Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., Xu, R., Zhu, Q., Ma, S., Wang, P., Bi, X., et al. Deepseek-r1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Han, Y., Huang, G., Song, S., Yang, L., Wang, H., and Wang, Y. Dynamic neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44 (11):7436–7456, 2021.

He, H., Wen, C.-K., Jin, S., and Li, G. Y. Deep learning-based channel estimation for beamspace mmwave massive MIMO systems. *IEEE Wireless Communications Letters*, 7(5):852–855, 2018.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

Hoydis, J., Cammerer, S., Aoudia, F. A., Vem, A., Binder, N., Marcus, G., and Keller, A. Sionna: An open-source library for next-generation physical layer research. *arXiv preprint arXiv:2203.11854*, 2022.

Hu, Q., Gao, F., Zhang, H., Jin, S., and Li, G. Y. Deep learning for channel estimation: Interpretation, performance, and comparison. *IEEE Transactions on Wireless Communications*, 20(4):2398–2412, 2021. doi: 10.1109/TWC.2020.3042074.

Huang, R., Zhu, S., Du, Y., and Zhao, H. Moe-loco: Mixture of experts for multitask locomotion. *arXiv preprint arXiv:2503.08564*, 2025.

Jacobs, R. A., Jordan, M. I., Nowlan, S. J., and Hinton, G. E. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991.

Kingma, D. and Ba, J. Adam: a method for stochastic optimization. In *International Conference on Learning Representations*, 2014.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Lepikhin, D., Lee, H., Xu, Y., Chen, D., Firat, O., Huang, Y., Krikun, M., Shazeer, N., and Chen, Z. Gshard: Scaling giant models with conditional computation and automatic sharding. In *International Conference on Learning Representations*.

Li, L., Chen, H., Chang, H.-H., and Liu, L. Deep residual learning meets OFDM channel estimation. *IEEE Wireless Communications Letters*, 9(5):615–618, 2020. doi: 10.1109/LWC.2019.2962796.

Liu, A., Feng, B., Xue, B., Wang, B., Wu, B., Lu, C., Zhao, C., Deng, C., Zhang, C., Ruan, C., et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

Luan, D. and Thompson, J. Achieving robust generalization for wireless channel estimation neural networks by designed training data. In *ICC 2023-IEEE International Conference on Communications*, pp. 3462–3467. IEEE, 2023.

Mao, H., Lu, H., Lu, Y., and Zhu, D. Roemnet: Robust meta learning based channel estimation in OFDM systems. In *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, pp. 1–6. IEEE, 2019.

Muqeeth, M., Liu, H., Liu, Y., and Raffel, C. Learning to route among specialized experts for zero-shot generalization. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 36829–36846, 2024.

Neumann, D., Wiese, T., and Utschick, W. Learning the MMSE channel estimator. *IEEE Transactions on Signal Processing*, 66(11):2905–2917, 2018.

O'Shea, T. and Hoydis, J. An introduction to deep learning for the physical layer. *IEEE Transactions on Cognitive Communications and Networking*, 3(4):563–575, 2017. doi: 10.1109/TCCN.2017.2758370.

Samuel, N., Diskin, T., and Wiesel, A. Deep MIMO detection. In *2017 IEEE 18th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pp. 1–5. IEEE, 2017.

Soltani, M., Pourahmadi, V., Mirzaei, A., and Sheikhzadeh, H. Deep learning-based channel estimation. *IEEE Communications Letters*, 23(4):652–655, 2019.

Wang, L., Gao, H., Zhao, C., Sun, X., and Dai, D. Auxiliary-loss-free load balancing strategy for mixture-of-experts. *arXiv preprint arXiv:2408.15664*, 2024.

Wen, C.-K., Shih, W.-T., and Jin, S. Deep learning for massive MIMO CSI feedback. *IEEE Wireless Communications Letters*, 7(5):748–751, 2018.

Ye, H., Li, G. Y., and Juang, B.-H. Power of deep learning for channel estimation and signal detection in OFDM systems. *IEEE Wireless Communications Letters*, 7(1): 114–117, 2017.