MoDE: A Mixture-of-Decision-Experts Reinforcement Learning Architecture for High-Dimensional Coupled Wireless Optimization

Shiyi Lin, Hongyang Du

Abstract—Next-generation wireless networks have garnered notable attention due to their ability to deliver ultra-reliable connectivity, extreme data rates, and low-latency services, significantly impacting applications ranging from immersive communications to intelligent transportation. However, achieving these ambitious targets is hindered by tightly coupled design variables across multiple layers of the system, give rise to high-dimensional, nonconvex optimization problems that are difficult to solve efficiently in real time. To address this challenge and maximize spectral efficiency (SE), we propose a novel learning-based framework that integrates Mixture-of-Decision-Experts (MoDE) with reinforcement learning (RL). We evaluate the proposed architecture against a heuristic baseline with low inference complexity and a conventional multilayer perceptron (MLP)-based deep reinforcement learning (DRL) approach. Numerical results show that our method achieves higher SE than the heuristic baseline and delivers both faster convergence and superior SE compared to the MLP-based approach, with allocating more experts to higher-dimensional layers further improving performance over uniform-expert-allocations design, demonstrating both efficiency and robustness.

Index Terms—Generative AI (GAI), networks

1 Introduction

Recently, wireless networks have been rapidly evolving towards sixth-generation (6G) systems, driven by demands for ubiquitous connectivity, extreme data rates, low-latency services, and dense device deployment [1], [2]. These trends introduce three interrelated characteristics. (i) Ultra-dense and heterogeneous infrastructure emerges[3], encompassing many base stations[4], reconfigurable intelligent surfaces(RIS) [5], unmanned aerial vehicles(UAVs) [6] and edge nodes [7]; (ii) massive device connectivity with diverse quality-of-service(QoS) requirements must be supported [8], [9]; and (iii) stringent performance targets are imposed, including high spectral and energy efficiency together with low latency [2], [10]. Collectively, these characteristics significantly increase the dimensionality and nonlinearity of system models[11], [12], thereby turning classical design tasks into large-scale, high-dimensional, and nonconvex optimization problems[13], [14]. These challenges manifest as tightly coupled decision problems. For example, antenna placement, beamforming and power allocation interacting through the physical channel and receiver processing across many wireless applications, where the optimization of one component strongly affects others[15]. Meanwhile, practical deployments further demand real-time or near real-time solutions, imposing severe computational pressure and requiring algorithms that deliver high-quality joint decisions within tight latency budgets[16], [17].

To fully realize the performance benefits of next-generation wireless systems, these coupled problems must be solved efficiently and robustly under realistic modeling uncertainty. Prior work has pursued model-based and learning-based approaches. For example, a hierarchical three-stage beam training scheme achieves low-overhead antenna position and beam alignment by using multi-resolution codebooks and coarse-to-fine search, but it depends on pre-computed codebooks and still requires exhaustive sampling of candidate positions within each codebook level, which limits adaptability in highly dynamic mobile environments where channel geometry and line-of-sight conditions change rapidly [18]. Another effective class of methods leverages fractional programming together with block coordinate descent to transform certain ratio-type objectives into tractable subproblems and to alternate closed-form updates [19], [20]. These approaches deliver strong performance for joint power control, beamforming, and scheduling, yet they rely on iterative subproblem solves and often require grid or combinatorial searches when antenna placement or discrete configuration decisions are involved, and their performance and runtime can be sensitive to initialization. As a result, they are not always suitable for real-time adaptation in large-scale heterogeneous deployments. As for learning-based approaches, for example, DRL circumvent explicit modeling by learning policies from interaction, and therefore have been applied to joint beamforming, power control, and interference coordination [21]. Nevertheless, DRL exhibits its own limitations, where many of its methods are sample-inefficient, struggle with very large action spaces, and exhibit unstable training when the control problem decomposes into multiple interacting subtasks. This contrast highlights a clear gap, that is, neither purely model-based nor vanilla end-to-end learning fully balances scalability, data efficiency, specialization across subtasks, and stable training in tightly coupled wireless control problems.

Taken together, these limitations motivate the following research question:

How can we obtain data-efficient, scalable policies that solve deeply coupled wireless optimization problems while avoiding interference between heterogeneous subtasks?

Addressing this question calls for architectures that (i) decompose complex decisions into specialized subtasks, (ii) increase representational capacity in a controlled way, and (iii) maintain stable learning for continuous, multimodal control. Mixture-of-experts (MoE) architectures are a natural candidate because they distribute capacity across specialized experts via learned routing, which reduces cross-task interference and enables more focused representation learning [22], [23]. Recent LLM work has explored dense activation or dense training variants of MoE as a practical way to stabilize expert initialization and encourage richer expert specialization before committing to sparse routing[24], [25]. Dense activation strategies that route to many or all experts improve specialization and reduce routing overlap at the cost of increased computation per example. This trade-off, that is, higher compute for stronger specialization and more robust learning, matches our goal of solving tightly-coupled wireless optimization tasks where representational fidelity and stable training matter.

To satisfy the requirements above, we propose a MoDE driven decision aided DRL framework tailored for tightly-coupled wireless optimization problems existing in multiple application domains. Specifically, MoDE has three design characteristics. (i) Task decomposition. The MoDE-actor is organized in layers that reflect natural groupings of the joint action such as position selection, power allocation, and beamforming decoding. Each layer produces the layer-specific action component and is modeled by its own MoDE block, which reduces the dimensionality and cross-talk facing any single module and clarifies learning targets for experts. (ii) Expert specialization. Rather than uniformly assigning the same number of experts to every layer, MoDE allows heterogeneous expert counts and internal segmentation: high-dimensional or more complex layers receive more experts, while simpler layers use fewer. We also design shared experts across layers to capture common structure. This allocation improves representational fidelity and stabilizes learning in composite action spaces. (iii) Adaptive routing and dense activation. MoDE allows routing policies to activate multiple experts when subtasks interact strongly, trading extra computation for improved representational fidelity and stable learning. Algorithmically, MoDE is unified and agnostic to deployment. The training loop is end-to-end and implemented with an off-policy, entropy-regularized backbone so that continuous, multimodal control benefits from robust exploration and sample reuse. At deployment time, MoDE is flexible because it can run entirely on a single platform or be partitioned across devices.

To demonstrate the practical relevance and generality of MoDE, we evaluate it on three representative classes of tightly-coupled wireless control problems:(i) the optimization of Pinching Antenna Systems(PASS) requiring joint position and beamforming design optimization; (ii) Multi-User Multiple-Input Multiple-Output (MU-MIMO) where power allocation and beamforming design are jointly optimized; and (iii) joint transmission and reflection optimization in an Intelligent Reflecting Surface (IRS)-aided MIMO system. Although these scenarios rely on different physical mechanisms, including reconfigurable antenna geometry, spatial multiplexing at arrays, and environment shaping via metasurfaces, they all produce the same kind of multivariable, tightly-coupled decision structure, yielding high-dimensional, nonconvex joint optimization objectives. It is precisely this class of structured, high-dimensional problems that the MoDE framework is designed to address.

The main contributions of this paper are summarized as follows.

- We propose the MoDE, a general-purpose MoE that can be readily integrated into various RL frameworks for structured decision making. MoDE factorizes tightly-coupled, high-dimensional, and strongly nonconvex optimization problems into staged expert modules, and supports heterogeneous expert allocation, fine-grained expert segmentation and shared experts, enabling targeted specialization across decisions.
- We realize the decision-making idea within a DRL framework by integrating MoDE with Soft Actor–Critic (SAC) algorithm. We further provide concrete, scenario-specific algorithmic instantiations for three representative wireless problems in PASS, MU–MIMO, and IRS-aided MIMO cases.
- We perform many experiments that validate effectiveness of MoDE. Compared to a heuristic baseline and standard MLP-based RL, MoDE attains higher SE and faster convergence after sufficient training in our testbed scenarios. We also show that allocating more experts to higher-dimensional layers yields additional gains over uniform expert allocation.

2 Related work

This section focuses on three classical, tightly-coupled wireless optimization scenarios to clarify the concrete technical challenges that motivate MoDE.

2.1 Pinching Antenna positions and beamforming design in PASS.

The placement of pinching elements and the digital beamformer jointly determine the baseband-equivalent channel matrix entries, including their phases [26]. Activating different pinching elements modifies propagation paths and line-of-sight components and therefore changes per-stream channel gains and phase alignments [27], [28]. Consequently, a beamformer that provides coherent combining and high array gain for one antenna configuration may suffer phase misalignment or reduced gain under another configuration. Hence, jointly optimizing antenna placement and beamformer design generally yields better SE and energy efficiency than optimizing them separately.

To make the coupling between the pinching configuration and the precoder $\mathbf{W}$ explicit, we start from the receiving of each user$k$

$$y_k = \mathbf{g}_k^T(\ell)\mathbf{W}\mathbf{s} + \varepsilon_k,$$

where $\mathbf{g}_k(\ell) = [g_{1,k}(\ell_1), \ldots, g_{M,k}(\ell_M)]^T$ represents the position-dependent channel vector, with each element $g_{m,k}(\ell_m)$ capturing the propagation characteristics from the m-th pinching element to user k. The resulting signal-to-interference-plus-noise ratio (SINR) becomes the following.

$$\mathrm{SINR}_k(\mathbf{W}, \ell) = \frac{|\mathbf{g}_k^T(\ell)\mathbf{w}_k|^2}{\sum_{j \neq k} |\mathbf{g}_k^T(\ell)\mathbf{w}_j|^2 + \sigma^2},$$

and the SE objective $R(\mathbf{W}, \ell)$ follows as

$$\text{maximize} \sum_k \lambda_k \log(1 + \mathrm{SINR}_k(\mathbf{W}, \ell)).$$

The critical observation lies in the multiplicative coupling within the terms $\mathbf{g}_k^T(\ell)\mathbf{w}_j$. This multiplicative structure renders the optimization landscape highly non-convex and prevents separable optimization.

Besides, circular dependency and mixed gradients exist. For any fixed antenna configuration ℓ, the optimal beamformer exhibits a regularized zero-forcing structure as

$$\mathbf{W}^*(\ell) = \left(\mathbf{G}^H(\ell)\mathbf{U}\mathbf{G}(\ell) + \frac{\sigma^2 \operatorname{tr}(\mathbf{U})}{P}\mathbf{I}\right)^{-1} \mathbf{G}^H(\ell)\mathbf{T},$$

which explicitly depends on the channel matrix $\mathbf{G}(\ell)$ determined by the pinching positions. This interdependence creates a circular dependency. Changes in antenna positions alter the optimal beamforming strategy, while adjustments in beamforming dictate new optimal antenna configurations. The coupling manifests concretely in the optimization gradients. Considering the partial derivative with respect to a position parameter ℓ_m as

$$\frac{\partial R}{\partial \ell_m} = \sum_k \frac{\partial R_k}{\partial \mathrm{SINR}_k} \cdot \frac{\partial \mathrm{SINR}_k(\mathbf{W}, \ell)}{\partial \ell_m},$$

where the gradient components inherently mix beamforming weights $\mathbf{W}$ with channel derivatives $\partial_{\ell_m}\mathbf{g}_k(\ell)$, further proving the coupling.

2.2 Power allocation and beamforming in MU-MIMO.

In MU-MIMO scenarios, system performance is shaped primarily by how transmit power is allocated across streams and by the selected beamforming vectors [29], [30]. Adjusting the power assigned to a stream directly changes the received signal strengths and alters the interference experienced by other users [31]. Modifying beamforming vectors changes the spatial distribution of radiated energy and thus reshapes per-user channel gains and interference geometry [32], [33]. Because power allocation and beamforming influence each other, an adjustment on one side typically changes the optimal choice on the other and the combined problem is high-dimensional and nonconvex. Therefore, treating power allocation and beamforming as a joint design problem yields better opportunities to balance throughput fairness and energy efficiency.

This coupling arises mathematically from the signal model where user k receives

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

where $\mathbf{h}_k \in \mathbb{C}^{N_t}$ denotes the channel vector to user k, $\mathbf{w}_k \in \mathbb{C}^{N_t}$ is the beamforming vector, and s_k is the transmitted symbol with $\mathbb{E}[|s_k|^2] = 1$. Let $\mathbf{w}_k = \sqrt{p_k}\mathbf{v}_k$, where $p_k \geq 0$ represents the power allocated to user k and $\mathbf{v}_k$ is the normalized beamforming direction as $\|\mathbf{v}_k\| = 1$. Substituting this decomposition yields

$$\mathrm{SINR}_k = \frac{p_k |\mathbf{h}_k^H \mathbf{v}_k|^2}{\sum_{j \neq k} p_j |\mathbf{h}_k^H \mathbf{v}_j|^2 + \sigma_k^2}.$$

This formulation reveals the multiplicative coupling. The SINR for each user depends on the product of its allocated power p_k and the beamforming gain $|\mathbf{h}_k^H \mathbf{v}_k|^2$, while also being affected by interference terms that similarly mix power allocations and beamforming directions from other users.

The coupling further manifests in the optimality conditions. For fixed beamforming directions $\{\mathbf{v}_k\}$, the power allocation problem becomes maximizing the SE objective R as

$$\underset{\{p_k\}}{\text{maximize}} \sum_{k=1}^{K} \log\left(1 + \frac{p_k |\mathbf{h}_k^H \mathbf{v}_k|^2}{\sum_{j \neq k} p_j |\mathbf{h}_k^H \mathbf{v}_j|^2 + \sigma_k^2}\right),$$

which is a non-convex problem due to the interference terms. Conversely, for fixed power allocation $\{p_k\}$, the optimal beamforming directions are given by the regularized zero-forcing solution

$$\mathbf{v}_k^\star \propto \left(\sum_{j=1}^{K} p_j \mathbf{h}_j \mathbf{h}_j^H + \mu \mathbf{I}\right)^{-1} \mathbf{h}_k,$$

where the beamforming for user k explicitly depends on all power allocations $\{p_j\}$ through the interference covariance matrix. This circular dependency creates a non-convex joint optimization landscape where adjustments in power allocation necessitate corresponding changes in beamforming strategy, and vice versa. The gradient of the SE with respect to power allocation further illustrates this coupling as

$$\frac{\partial R}{\partial p_k} = \sum_{i=1}^{K} \frac{\partial R_i}{\partial \mathrm{SINR}_i} \cdot \frac{\partial \mathrm{SINR}_i}{\partial p_k},$$

where each term $\partial \mathrm{SINR}_i / \partial p_k$ involves products of power variables and beamforming-dependent channel gains.

2.3 IRS phase and beamforming design in IRS-aided MIMO.

The configuration of IRS phase shifts and transmit beamforming collectively shapes the end-to-end amplitude and phase response of wireless channels [5]. On one hand, tuning the IRS phase shifts modifies the composite channel response by applying a multiplicative effect to signals along reflected paths. On the other hand, adapting the transmit beamforming alters the spatial excitation of the propagation environment, influencing how signals combine after reflection [34], [35]. Due to the fact that IRS phases and transmitter beamforming interact multiplicatively and that IRS elements have unit modulus constraints, changes on one side change the best choice on the other and the joint problem is strongly coupled and nonconvex [36], [37]. As a result, joint optimization of IRS phases and transmitter beamforming is required to reach favorable trade-offs in SE and robustness.

Consider an IRS-aided MIMO system where the base station (BS) employs N_t antennas, the IRS contains M reflecting elements, and K single-antenna users are served. The overall channel to user k comprises both the direct BS-user link $\mathbf{h}_{d,k}^H$ and the cascaded BS-IRS-user link as

$$\mathbf{h}_k^H(\boldsymbol{\Theta}) = \mathbf{h}_{d,k}^H + \mathbf{h}_{r,k}^H \boldsymbol{\Theta} \mathbf{G},$$

where $\mathbf{G}$ denotes the BS-IRS channel, $\mathbf{h}_{r,k}$ represents the IRS-user channel, and $\boldsymbol{\Theta} = \mathrm{diag}(\beta_1 e^{j\theta_1}, \ldots, \beta_M e^{j\theta_M})$ is the IRS reflection matrix with phase shifts $\theta_m \in [0, 2\pi)$ and reflection amplitudes $\beta_m \in [0,1]$. The received signal at user k becomes

$$y_k = \mathbf{h}_k^H(\boldsymbol{\Theta}) \sum_{j=1}^{K} \mathbf{w}_j s_j + n_k,$$

where $\mathbf{w}_j \in \mathbb{C}^{N_t}$ is the beamforming vector for user j. The coupling mechanism becomes explicit when we isolate the IRS phase contribution. Let $\mathbf{v} = [e^{j\theta_1}, \ldots, e^{j\theta_M}]^H$ represent the IRS phase shift vector, and define the effective cascaded channel for user k as $\boldsymbol{\Phi}_k = \mathrm{diag}(\mathbf{h}_{r,k}^H)\mathbf{G}$. The composite channel can be rewritten as

$$\mathbf{h}_k^H(\mathbf{v}) = \mathbf{h}_{d,k}^H + \mathbf{v}^H \boldsymbol{\Phi}_k.$$

Substituting into the SINR expression reveals the multiplicative coupling as

$$\mathrm{SINR}_k = \frac{|(\mathbf{h}_{d,k}^H + \mathbf{v}^H \boldsymbol{\Phi}_k)\mathbf{w}_k|^2}{\sum_{j \neq k} |(\mathbf{h}_{d,k}^H + \mathbf{v}^H \boldsymbol{\Phi}_k)\mathbf{w}_j|^2 + \sigma_k^2}.$$

This formulation demonstrates that both the desired signal power and interference terms involve products between the IRS phase vector $\mathbf{v}$ and the beamforming matrix $\mathbf{W}$, creating a bilinear coupling structure.

The non-convexity is further exacerbated by the unit modulus constraints $|v_m| = 1$ on IRS elements. For fixed beamforming $\mathbf{W}$, the IRS optimization problem becomes maximizing the SE R as

$$\underset{\mathbf{v}}{\text{maximize}} \sum_{k=1}^{K} \log\left(1 + \mathrm{SINR}_k(\mathbf{v})\right),$$

which is hard due to the non-convex objective and constraints. Conversely, for fixed IRS configuration $\mathbf{v}$, the optimal beamforming follows the MMSE structure as

$$\mathbf{w}_k^\star \propto \left(\sum_{j=1}^{K} \mathbf{h}_j(\mathbf{v}) \mathbf{h}_j^H(\mathbf{v}) + \mu \mathbf{I} \right)^{-1} \mathbf{h}_k(\mathbf{v}),$$

where the beamforming solution explicitly depends on the IRS-dependent composite channels.

Besides, the gradient analysis further illuminates the coupling. Considering the partial derivative of SE with respect to an IRS phase parameter θ_m as

$$\frac{\partial R}{\partial \theta_m} = \sum_{k=1}^{K} \frac{\partial R_k}{\partial \mathrm{SINR}_k} \cdot \frac{\partial \mathrm{SINR}_k}{\partial \theta_m},$$

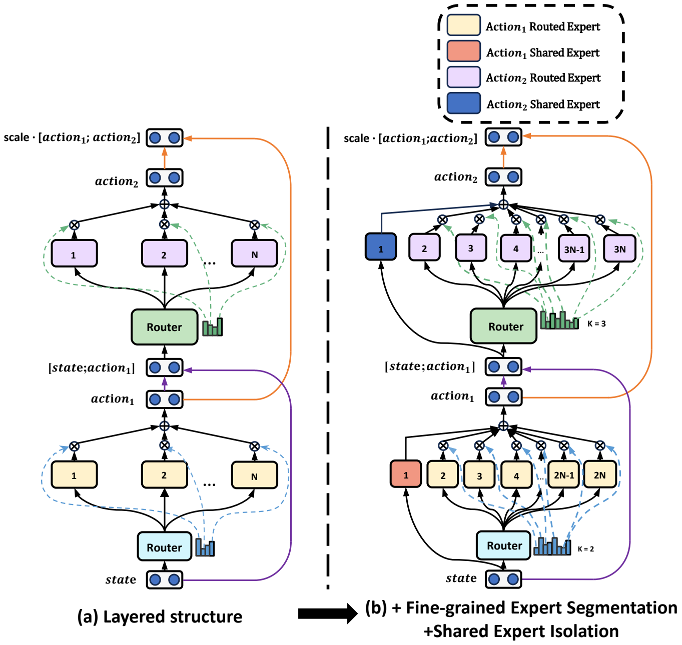
where each term $\partial \mathrm{SINR}_k / \partial \theta_m$ involves intricate products of beamforming vectors, channel matrices, and other IRS phase parameters. This mathematical structure confirms that IRS phase shifts and beamforming vectors are intrinsically coupled through multiplicative relationships in both signal and interference components, requiring joint optimization approaches to achieve optimal performance.

3 MoDE-aided-DRL

In this section, we propose the MoDE-aided DRL method. Specifically, we first discuss the MoDE architecture, and then we explain how to use the large language model(LLM)-based agent as evaluators to feedback the sum SE as the reward for DRL algorithms. We also provide a theoretical complexity analysis together with concrete numeric measurements using the PASS case. The proposed MoDE-SAC algorithm is designed as a general policy network intended for a broad class of wireless optimization problems characterized by multi-layer coupling, high dimensionality, and strong non-convexity. In this work, we specialize MoDE on the three representative coupled-decision scenarios from Section 2 to demonstrate the scalability of the framework.

3.1 MoDEActor Architecture

Our policy network is a two-stage MoDE that jointly predicts a configuration action and a high-dimensional signal action. As shown in Fig. 1, the MoDE Actor consists of:



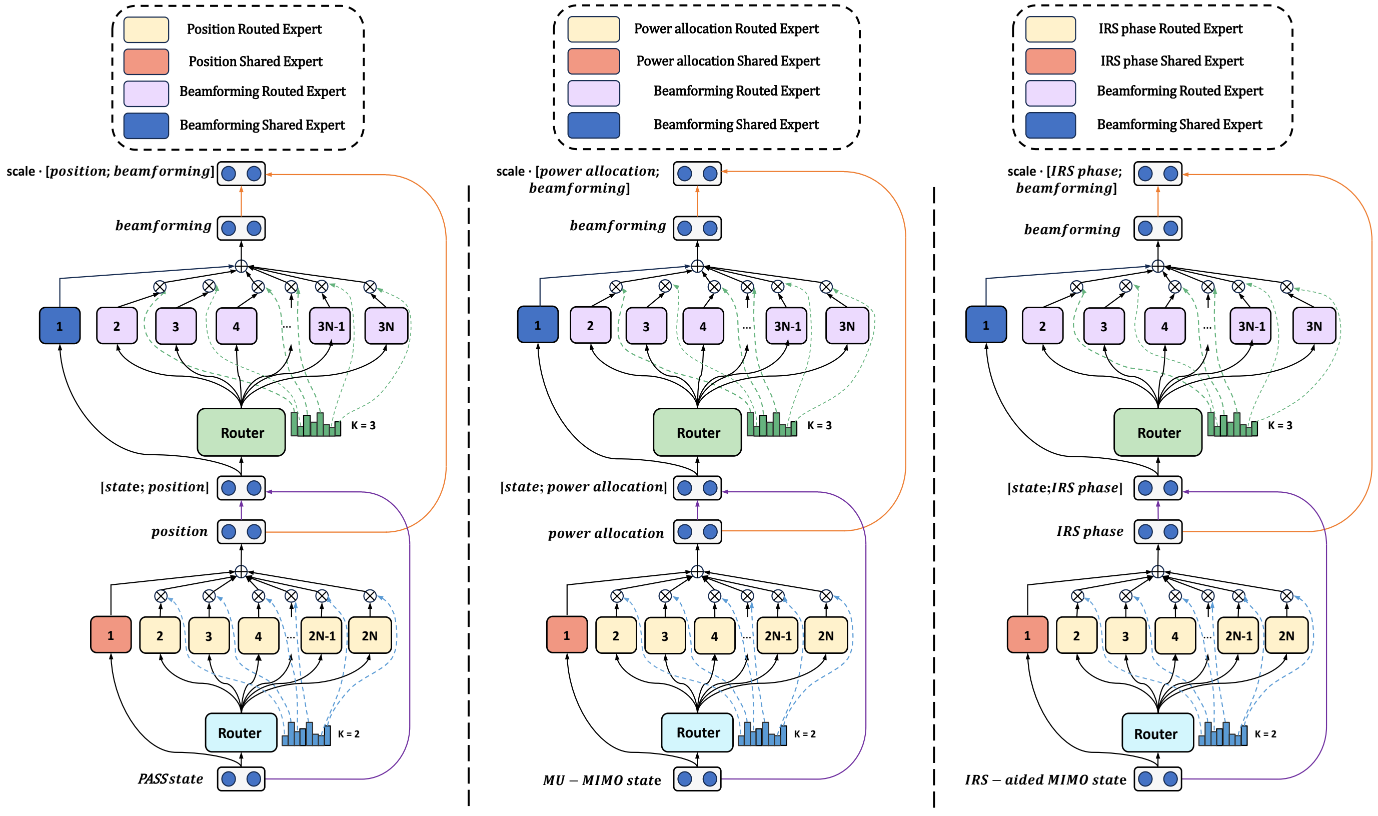
(a) Layered structure

(b) + Fine-grained Expert Segmentation +Shared Expert Isolation

Fig. 1. Illustration of general MoDE. Subfigure (a) showcases the layered MoDE structure. Subfigure (b) illustrates the fine-grained expert segmentation strategy and the integration of the shared expert isolation strategy.

3.1.1 *Layered MoDE Architecture with Heterogeneous Expert Allocation*

In a conventional MoE actor, all experts receive the same state input and are jointly trained to output actions for a


Position Routed Expert
Position Shared Expert
Beamforming Routed Expert
Beamforming Shared Expert
scale · [position; beamforming]
beamforming
1
2
3
4
3N-1
3N
Router
K = 3
[state; position]
position
1
2
3
4
2N-1
2N
Router
K = 2
PASSstate
Power allocation Routed Expert
Power allocation Shared Expert
Beamforming Routed Expert
Beamforming Shared Expert
scale · [power allocation; beamforming]
beamforming
Router
K = 3
[state; power allocation]
power allocation
Router
K = 2
MU − MIMO state
IRS phase Routed Expert
IRS phase Shared Expert
Beamforming Routed Expert
Beamforming Shared Expert
scale · [IRS phase; beamforming]
beamforming
Router
K = 3
[state;IRS phase]
IRS phase
Router
K = 2
IRS − aided MIMO state

(a) MoDE in PASS case **(b) MoDE in MU-MIMO case** **(c) MoDE in IRS-aided MIMO case**

Fig. 2. Illustration of MoDE for PASS, MU-MIMO, and IRS-aided-MIMO.

single decision step. However, direct application creates two main issues in the application of high-dimensional coupled wireless optimization:

- Coupled learning difficulty. Structural/configuration decisions and signal-level actions require different feature sensitivities and inductive biases. Training a single-layer MoE to handle both kinds of outputs induces noisy gradient updates and routing conflicts as configuration-related gradients interfere with signal-level specialization.
- Slow convergence and exploration cost. When the action space is large and multi-stage, the MoDE router simultaneously learn to route inputs for both tasks, which increases exploration cost and delays expert specialization.

If we separate the decision process into multiple MoDE layers, each layer can specialize in its own subtask. This "layered" design transforms the learning problem from one large entangled decision into two simpler sequential decisions. This hierarchical decomposition maintains the computational efficiency of standard MoDE and achieving higher specialization without interference from unrelated subtasks. In pursuit of the goal, we design a Layered MoDE Actor consisting of two sequential MoDE stages:

- Configuration MoDE layer: Routes the state to configuration experts, producing a structural/configuration action.
- Signal MoDE layer: Takes both the state and the chosen configuration as input, which we express as an enriched state

$$\tilde{s} = [s; a] \in \mathbb{R}^{d_{\text{state}}+d_{\text{config}}}, \qquad (1)$$

where a denotes the intermediate configuration action and d_{config} represents its dimensionality. Then we can rout the enriched state $\tilde{s}$ to experts in the signal MoDE layer for the final optimization.

Crucially, we assign different numbers of experts to the two MoDE layers. The configuration output is typically correspondingly low and its mapping from state to antenna coordinates is relatively simple, while the signal mapping is high dimensional and highly nonlinear. Allocating more experts and finer grained segmentation to the signal layer increases representational capacity where it is most needed and enables distinct experts to specialize on diverse channel configurations. At the same time, assigning a smaller expert set to the configuration layer avoids unnecessary overparameterization and reduces routing overhead for the simpler subtask.

Fig. 2 shows how the same general two-stage MoDE can be specialized to three representative wireless application domains. In all three cases, the Signal Layer corresponds to final beamforming design while the Configuration Layer is instantiated differently depending on the scenario. In PASS, the layer-1 configuration is the pinching-antenna position selection; for MU–MIMO the layer-1 configuration corresponds to user-level power allocation; and for IRS-aided MIMO the layer-1 configuration is the IRS passive phase-shift pattern. Presenting these mappings explicitly clarifies how the layered MoDE acts as a general architecture. The same staged decomposition and heterogeneous expert allocation can be instantiated per scenario by changing the layer-semantics while keeping the overall training and deployment pipeline intact.

3.1.2 *Fine-Grained Expert Segmentation and Shared Experts Isolation*

In conventional MoDE with limited experts, a single expert often has to capture highly diverse patterns of knowledge. This forces the expert parameters to capture incompatible patterns, reducing their ability to optimize for any one pattern[23]. To address this, one strategy is to split an expert into smaller subexperts and activate more of them simultaneously. In this way, diverse knowledge can be distributed across specialized subspaces, allowing each subexpert to focus on a narrower scope of representation learning [38], [39].

However, diversity alone is not sufficient. In many routing strategies, different experts inadvertently learn overlapping common knowledge, leading to redundancy and inefficient parameter utilization. This overlap reduces the capacity available for capturing unique, specialized features. To mitigate this, we introduce dedicated shared experts that always receive input. These shared experts act as repositories of common knowledge, thereby freeing the fine-grained experts to focus exclusively on specialized features [39]. Together, segmentation and isolation create a complementary mechanism,that is, segmentation enhances specialization, while shared experts centralize generalization.

Building on the Layered MoDE Actor architecture described above, we implement two categories of experts, configuration experts and signal experts. Each category contains fine-grained experts and one shared expert. Every expert is parameterized as a two-hidden-layer fully connected network with separate output heads for the mean and log standard deviation of a Gaussian distribution. The feedforward computation can be expressed as:

$$h_1 = \mathrm{ReLU}(W_1 x + b_1), \quad W_1 \in \mathbb{R}^{d_h \times d_{\text{in}}}, \tag{2}$$

$$h_2 = \mathrm{ReLU}(W_2 h_1 + b_2), \quad W_2 \in \mathbb{R}^{d_h \times d_h}, \tag{3}$$

$$\mu = W_\mu h_2 + b_\mu, \quad W_\mu \in \mathbb{R}^{d_{\text{out}} \times d_h}, \tag{4}$$

$$log\sigma = W_\sigma h_2 + b_\sigma, \quad W_\sigma \in \mathbb{R}^{d_{\text{out}} \times d_h}, \tag{5}$$

where $x \in \mathbb{R}^{d_{\text{in}}}$ is the input feature vector, d_{in} is the input dimension, d_h is the hidden dimension, and d_{out} is the output dimension. The parameters $W_1, W_2, W_\mu, W_\sigma$ are trainable weight matrices, and $b_1, b_2, b_\mu, b_\sigma$ are their associated bias vectors. The outputs $\mu \in \mathbb{R}^{d_{\text{out}}}$ and $\log\sigma \in \mathbb{R}^{d_{\text{out}}}$ represent the mean and log standard deviation of the Gaussian distribution, respectively.

The output of one expert feedforward network is

$$\mathrm{ExpertFFN}(x) = (\mu, \log\sigma).$$

Hence, Fine-Grained Experts outputs and Shared expert outputs can be expressed as:

$$(\mu_i^{\text{config}}, \log\sigma_i^{\text{config}}) = E_i^{\text{config}}(s), \quad i = 1 \dots n, \tag{6}$$

$$(\mu_j^{\text{signal}}, \log\sigma_j^{\text{signal}}) = E_j^{\text{signal}}(\tilde{s}), \quad j = 1 \dots m, \tag{7}$$

$$(\mu_k^{\text{config}}, \log\sigma_k^{\text{config}}) = E_k^{\text{config}}(s), \tag{8}$$

$$(\mu_k^{\text{signal}}, \log\sigma_k^{\text{signal}}) = E_k^{\text{signal}}(\tilde{s}), \tag{9}$$

where n and m denote the segmentation factors for the configuration and signal experts, respectively. Specifically, each configuration expert FFN is split into n sub-experts by reducing the hidden dimension of each FFN to $\frac{1}{n}$ of its original size. To maintain constant computational cost, the number of simultaneously activated sub-experts is increased to n times the original. An analogous segmentation is applied to the signal experts with factor m.

Each MoDE layer employs a lightweight router that maps the layer input x to per-expert gating logits. The logits are converted to a normalized weight vector by a softmax as

$$\mathbf{w} = \frac{\exp((R\mathbf{x} + \mathbf{r})_i)}{\sum_{j=1}^{E} \exp((R\mathbf{x} + \mathbf{r})_j)} \in \mathbb{R}^E,$$

where $R \in \mathbb{R}^{E \times d_{\text{in}}}$ and $\mathbf{r} \in \mathbb{R}^E$ are the router parameters, d_{in} is the input dimension, and E is the number of fine-grained experts in that layer. The resulting weights satisfy $w_i \geq 0$ and $\sum_{i=1}^{E} w_i = 1$ for each sample. Using the per-sample weights w_i, the mixed distribution parameters for the configuration layer are computed by adding the shared expert output and the weighted sum of fine experts:

$$\mu^{\text{config}}(s) = \mu_{\text{shared}}^{\text{config}}(s) + \sum_{i=1}^{n_{\text{fine}}} w_i^{\text{config}}(s) \mu_i^{\text{config}}(s), \tag{10}$$

$$\log\sigma^{\text{config}}(s) = \log\sigma_{\text{shared}}^{\text{config}}(s) + \sum_{i=1}^{n_{\text{fine}}} w_i^{\text{config}}(s) \log\sigma_i^{\text{config}}(s). \tag{11}$$

We sample the intermediate configuration action as

$$a_1 = \tanh\left(\mu^{\text{config}}(s) + \sigma^{\text{config}}(s) \cdot \epsilon\right), \quad \epsilon \sim \mathcal{N}(0, I). \tag{12}$$

Here, $a_1 \in \mathbb{R}^{d_{\text{config}}}$ represents the configuration decision. For the signal control stage, we enrich the state with the intermediate action a_1, forming the augmented input as equation (1). The signal action distribution parameters are computed similarly:

$$\mu^{\text{signal}}(\tilde{s}) = \mu_{\text{shared}}^{\text{signal}}(\tilde{s}) + \sum_{j=1}^{n_{\text{fine}}} w_j^{\text{signal}}(\tilde{s}) \mu_j^{\text{signal}}(\tilde{s}), \tag{13}$$

$$\log\sigma^{\text{signal}}(\tilde{s}) = \log\sigma_{\text{shared}}^{\text{signal}}(\tilde{s}) + \sum_{j=1}^{n_{\text{fine}}} w_j^{\text{signal}}(\tilde{s}) \log\sigma_j^{\text{signal}}(\tilde{s}). \tag{14}$$

We then sample the signal action as

$$a_2 = \tanh\left(\mu^{\text{signal}}(\tilde{s}) + \sigma^{\text{signal}}(\tilde{s}) \cdot \epsilon'\right), \quad \epsilon' \sim \mathcal{N}(0, I). \tag{15}$$

Here, $a_2 \in \mathbb{R}^{d_{\text{signal}}}$ represents the signal action.

The final action vector a is the concatenation of configuration and signal actions as

$$a = \text{scale} \cdot [a_1 \ ; \ a_2] + \text{bias}. \tag{16}$$

3.1.3 Load Balance Consideration

Automatically learned routing strategies may encounter the issue of load imbalance, which manifests two notable defects. Firstly, there is a risk of routing collapse. The router repeatedly selects only a small subset of experts for most tokens, preventing other experts from sufficient training [39]. Secondly, persistent load imbalance prevents fair utilization of all experts, which can lead to unequal convergence rates and wasted model capacity.

Diversity loss. To mitigate routing collapse, we introduce a diversity loss that encourages the router to maintain high-entropy routing distributions. This ensures that tokens are spread more evenly across experts rather than collapsing onto a few dominant ones:

$$\bar{H}_{\text{config}} = \frac{1}{B} \sum_{b=1}^{B} \left[-\sum_{i=1}^{E} w_{b,i}^{\text{config}} \log w_{b,i}^{\text{config}} \right], \tag{17}$$

$$\bar{H}_{\text{signal}} = \frac{1}{B} \sum_{b=1}^{B} \left[-\sum_{i=1}^{E} w_{b,i}^{\text{signal}} \log w_{b,i}^{\text{signal}} \right], \tag{18}$$

$$\mathcal{L}_{\text{diversity}} = -\left(\bar{H}_{\text{config}} + \bar{H}_{\text{signal}} \right), \tag{19}$$

where B represents the batch size, E represents the number of experts in a router and $w_{b,i}^{\text{config}}$ is normalized routing weight of configuration layer MoDE for sample b and expert i while $w_{b,i}^{\text{signal}}$ is normalized routing weight of signal layer MoDE for sample b and expert i. This term is minimized when the router assigns weights more evenly within each sample, thus reducing the chance of collapse to one or two experts.

Balance loss. In addition, we define an expert-level balance loss that directly regulates the long-term average usage of experts across a batch. This term penalizes disproportionate routing and promotes equalized expert utilization, as expressed by:

$$\mu_i^{\text{config}} = \frac{1}{B} \sum_{b=1}^{B} w_{b,i}^{\text{config}}, \quad \mu_i^{\text{signal}} = \frac{1}{B} \sum_{b=1}^{B} w_{b,i}^{\text{signal}}, \tag{20}$$

$$\mathcal{L}_{\text{balance}} = \frac{1}{E} \sum_{i=1}^{E} \left(\mu_i^{\text{config}} \right)^2 + \frac{1}{E} \sum_{i=1}^{E} \left(\mu_i^{\text{signal}} \right)^2. \tag{21}$$

This term encourages average usage to be spread across experts. This loss reaches its minimum when all experts are equally utilized on average, i.e.,when $\mu_i = \frac{1}{E}$ for every expert.

We combine diversity loss and balance loss into a single expert loss term that softly guides the router towards balanced yet specialized routing, with $\lambda = 10^{-3}$ in our experiments, balancing the stronger anti-collapse effect with a softer global balancing term.

$$\mathcal{L}_{\text{expert}} = \underbrace{\mathcal{L}_{\text{diversity}}}_{\text{anti-collapse}} + \lambda \underbrace{\mathcal{L}_{\text{balance}}}_{\text{load equalization}}. \tag{22}$$

3.2 DRL with SAC feedback Framework

SAC is explicitly formulated for reinforcement-learning tasks with continuous actions and employs entropy-regularized policy optimization to promote robust exploration, while its off-policy training reuses prior experience to improve data efficiency[40]. These properties make SAC a strong fit for continuous placement control under time-varying radio environments, enabling adaptive, sample-efficient updates[41]. For these reasons we adopt SAC as the RL backbone and integrate it with a layered MoDE actor in this work to provide large representational capacity while preserving training stability. The general algorithm for implementing SAC is shown as Algorithm 1. Specifically, the management model initializes with parameters θ, ϕ, ψ and is trained through interactions with K LLM-based agents that simulate user feedback. Training continues for E episodes; each episode e starts from an initial state s and iterates until a terminal condition is met. A terminal condition indicates episode completion and may correspond to task success, a maximum step budget, or a failure event. At each decision step, LLM-based agents sample an action a from the policy $\pi_\theta(s)$ and receive rewards formed by assessments, where each agent_k outputs a user-centric utility,that is, per-step SE. State transitions (s, a, r, s') are stored in an off-policy replay buffer and sampled to perform gradient updates of the policy and critics. The LLM-based agent learns by minimizing an entropy-regularized objective, which promotes stable learning and robust exploration in continuous action spaces. Through iterative training over E episodes, the management model progressively refines its decision-making capability, leveraging simulated user feedback to converge toward a robustly trained management model.

Algorithm 1 Deep Reinforcement Learning using Soft Actor-Critic

```
Initialize: Actor–critic networks with parameters $\theta, \phi, \psi$;
            replay buffer $\mathcal{B}$; SAC empowered LLM-based
            agents $K$ to simulate $K$ users
Output: Trained actor–critic networks $\theta, \phi, \psi$
for each episode $e = 1, 2, \ldots, E$ do
    Initialize state $s$ while $s$ is not terminal do
        Generate action $a \leftarrow \pi_\theta(s)$
        Obtain reward $r \leftarrow \sum_{k=1}^{K} \text{agent}_k(s, a)$
        Observe next state $s'$
        Store transition $(s, a, r, s')$ in replay buffer $\mathcal{B}$
        Sample a minibatch of transitions from $\mathcal{B}$
        Update critic parameters $\phi, \psi$ by minimizing the soft
         Bellman residual
        Update policy parameters $\theta$ via gradient ascent on
         the expected soft value
        Adjust temperature parameter $\alpha$ to match the target
         entropy
        $s \leftarrow s'$
```

3.3 Overall Complexity

3.3.1 Complexity analysis setup

We adopt the following notation used throughout the complexity analysis in table 1, where $H_{e,p} = H_{e,b} = H/2$,

$K_p = a \cdot c$, and $K_b = a \cdot b$. We analyze complexity in two complementary ways.

TABLE 1
Notation and parameter definitions.

Parameter	Type	Meaning
a	Hyperparameter	Base number of experts
b	Hyperparameter	Beamforming segmentation factor
c	Hyperparameter	Position segmentation factor
O	State	Observation dimension
P	Action	Position action dimension
B	Action	Beamforming action dimension
H	Model parameter	Shared hidden size.
$H_{e,p}$	Model parameter	Position expert hidden size
$H_{e,b}$	Model parameter	Beamforming expert hidden size
K_p	Model parameter	Number of position experts
K_b	Model parameter	Number of beamforming experts
H_{router}	Model parameter	Router internal hidden size

TABLE 2
Comparison of concrete complexity for the PASS case using MoDE and MLP actor.

Component	Parameters	Forward FLOPs (approx)
MoDE actor		
Position expert (per)	29,708	—
Position experts (total)	118,832	—
Beamforming expert (per)	33,056	—
Beamforming experts (total)	198,336	—
Shared position expert	92,172	—
Shared beamforming expert	98,848	—
Routers (pos / bf)	24,324 / 26,374	—
MoDE total	**558,886**	**1,109,232**
Baseline MLP actor		
MLP: fc1 (O → H), fc2 (H → H)	46,592 / 262,656	—
MLP: mean / logstd heads	11,286 / 11,286	—
MLP total (H=512)	**331,820**	**661,504**

- **Parameter count.** Below we give exact layerwise sum expressions for the two main model families considered. (i) The per-expert feed-forward network (ExpertFFN) described in Sec. 3.1.2 is used in our MoDE actor. (ii) We use a two-layer MLP actor for comparison. As each ExpertFFN consists of two dense internal linear layers plus two output heads including mean and log-std. Denoting the expert input dimension by I, the expert hidden width by H, and the expert output dimension by O, the exact parameter count of a single expert is

$$\begin{aligned}\text{Params}_{\text{expert}}(I, H, O) = &\big(I \cdot H + H\big)\\ &+ \big(H \cdot H + H\big)\\ &+ \big(H \cdot O + O\big)\\ &+ \big(H \cdot O + O\big)\end{aligned}$$

Using the expert formula above, plus exact sums for shared experts and router layers, the dense MoDE parameter total is given by

$$\begin{aligned}\text{Params}_{\text{MoDE}} = &K_p \cdot \text{Params}_{\text{expert}}(O, H_{e,p}, P)\\ &+ K_b \cdot \text{Params}_{\text{expert}}(O + P, H_{e,b}, B)\\ &+ \text{Params}_{\text{shared\_pos}}(O, H, P)\\ &+ \text{Params}_{\text{shared\_bf}}(O + P, H, B)\\ &+ \text{Params}_{\text{routers}}.\end{aligned}$$

Asymptotically the dominant terms are those proportional to H^2 coming from multiple experts as

$$\text{Params}_{\text{MoDE}} = \Theta\big(a \cdot (OH + H(P + B) + H^2(\tfrac{1}{b} + \tfrac{1}{c})) + H^2\big).$$

For the two-layer MLP actor with hidden width H_{MLP}, input dimension O and total action dimension $A = P + B$, the exact parameter count is

$$\begin{aligned}\text{Params}_{\text{MLP}} = &\big(O \cdot H_{\text{MLP}} + H_{\text{MLP}}\big)\\ &+ \big(H_{\text{MLP}} \cdot H_{\text{MLP}} + H_{\text{MLP}}\big)\\ &+ \big(H_{\text{MLP}} \cdot A + A\big)\\ &+ \big(H_{\text{MLP}} \cdot A + A\big).\end{aligned}$$

Asymptotically the parameter count is

$$\text{Params}_{\text{MLP}} = \Theta(H_{\text{MLP}}^2 + H_{\text{MLP}}(O + A)).$$

- **FLOPs Analysis.** We analyze computational complexity using the standard multiply-accumulate operation convention, where each linear layer operation $\text{Linear}_{\text{in}\to\text{out}}$ requires exactly $2 \cdot \text{in} \cdot \text{out}$ FLOPs. This count includes the bias terms. Pointwise nonlinearities (e.g., ReLU) are treated as lower-order contributions and omitted from the leading-order analysis.

For the ExpertFFN block comprising two hidden linear layers and two parallel output heads, the FLOPs can be expressed as

$$\text{FLOPs}_{\text{expert}}(\text{in}, H, \text{out}) = 2\big(\text{in} \cdot H + H^2 + 2H \cdot \text{out}\big).$$

As for MoDE, we sum across all expert networks, shared components, and routing mechanisms as

$$\begin{aligned}\text{FLOPs}_{\text{MoDE}} = &K_p \cdot \text{FLOPs}_{\text{expert}}(O, H_{e,p}, P)\\ &+ K_b \cdot \text{FLOPs}_{\text{expert}}(O + P, H_{e,b}, B)\\ &+ \ \text{FLOPs}_{\text{shared\_pos}} + \text{FLOPs}_{\text{shared\_bf}},\\ &+ \text{FLOPs}_{\text{routers}} + \text{FLOPs}_{\text{aggregation}},\end{aligned}$$

where the aggregation FLOPs account for the weighted summation of expert outputs including mean and log-std, calculated as $2 \cdot (K_p \cdot P + K_b \cdot B)$. As for the two-layer MLP, FLOPs are summed as

$$\text{FLOPs}_{\text{MLP}} = 2\big(O \cdot H_{\text{MLP}} + H_{\text{MLP}}^2 + 2H_{\text{MLP}} \cdot A\big).$$

3.3.2 Concrete numeric example using PASS case

We now instantiate the algebraic expressions for the PASS environment used in the experiments with $N = 3$ waveguides, $M = 2$ pinching antennas per waveguide, and $K = 6$ users, which gives the following dimensions $O = 90, P = 6, B = 16, A = 22$. We set the model hyperparameters as $H = 256, H_{MLP} = 512, a = 2, b = 3, c = 2$.

From the instrumented numbers in Table 2, the MoDE actor uses approximately 5.59×10^5 parameters and 1.11×10^6 forward FLOPs at initialization, while the MLP with $H_{\text{MLP}} = 512$ uses approximately 3.32×10^5 parameters and 6.62×10^5 FLOPs. Thus, for this configuration, MoDE increases both parameter count and forward FLOPs by approximately a factor of

$$\frac{\text{Params}_{\text{MoDE}}}{\text{Params}_{\text{MLP}}} \approx 1.68, \qquad \frac{\text{FLOPs}_{\text{MoDE}}}{\text{FLOPs}_{\text{MLP}}} \approx 1.68.$$

Despite an approximately 1.68 times parameter and FLOPs increase, the MoDE is compatible with DRL practice and does not invalidate convergence behavior observed for typical DRL algorithms, which are proved in the experiments of Section 4.

4 Cases

In this section we describe the three representative application scenarios and explain how the proposed layered MoDE actor and the SAC backbone can be applied in each case. We perform experiments for the coupled decision-making problem in PASS and outline how the same architecture generalizes to MU-MIMO and IRS-aided-MIMO.

4.1 Joint decision-making problem in PASS

In this section we evaluate the proposed MoDE-aided SAC LLM-based agent on the pinching-antenna placement and beamforming decision problem and then show its effectiveness.

4.1.1 User-centric Spectral-Efficiency Maximization Problem

We consider a downlink multi-user scenario as Fig. 3 with N parallel waveguides at the base station (BS) and K single-antenna users deployed in a square region of side length D. Each waveguide carries M pinching antennas positioned along its length; the x-coordinate of the m-th antenna on waveguide n is denoted $x^p_{m,n} \in [0, D]$ and the waveguide height is d.

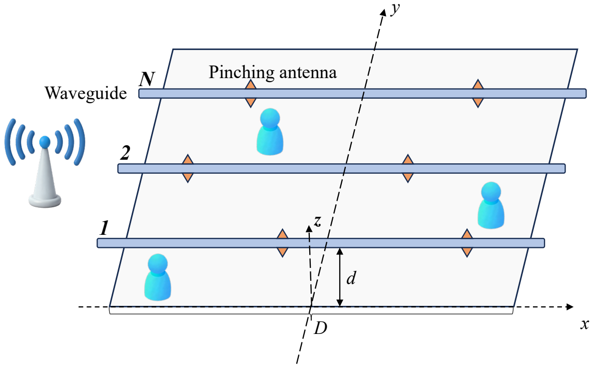


Fig. 3. PASS system model

The objective is user-centric in the sense that the LLM-based agent seeks to maximize an aggregate utility of the instantaneous per-user spectral efficiencies. Let $R_k(s, a)$ denote the instantaneous SE of user k under state s and action a. The optimization objective at each decision step is

$$\max_{a} \sum_{k=1}^{K} R_k(s, a). \tag{23}$$

The physical propagation model used in the environment follows the structure in PASS. The complex channel from all pinching antennas to user k is collected in the row vector $\mathbf{h}_k^H \in \mathbb{C}^{1\times NM}$, whose entries are small-scale line-of-sight gains. For the m-th antenna on waveguide n the entry is

$$h_{n,m,k} = \frac{\sqrt{\eta}\,\exp\big(-j\frac{2\pi}{\lambda}\|\psi_k - \psi^p_{m,n}\|\big)}{\|\psi_k - \psi^p_{m,n}\|},$$

where $\psi_k = (x_k, y_k, 0)$ is the user location, $\psi^p_{m,n} = (x^p_{m,n}, y^p_n, d)$ is the antenna location, $\lambda = \frac{c}{f_c}$ is the free-space wavelength, and $\eta = \frac{c}{2\pi f_c}$ is the propagation constant used in our implementation. The feed-point phase shifts along each waveguide are collected into a per-waveguide vector $\mathbf{g}_n$, and the block-diagonal pinching matrix $\mathbf{G} = \mathrm{blockdiag}(\mathbf{g}_1, \ldots, \mathbf{g}_N)$ captures the per-antenna phase reweighting induced by the waveguide feed and the specific pinching positions.

Given a baseband precoder matrix $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_K] \in \mathbb{C}^{NM\times K}$ and transmitted symbols x_k, the signal transmitted toward user k is $\mathbf{s}_k = \mathbf{G}\,\mathbf{w}_k\,x_k$. The received scalar signal at user k is therefore

$$y_k = \mathbf{h}_k^H \mathbf{s}_k + \sum_{j\neq k} \mathbf{h}_k^H \mathbf{s}_j + n_k,$$

where $n_k \sim \mathcal{CN}(0, \sigma_0^2)$ is complex Gaussian noise with variance σ_0^2.

From this model the instantaneous signal-to-interference-plus-noise ratio (SINR) for user k is

$$\mathrm{SINR}_k(\Phi^p, \mathbf{W}) = \frac{|\mathbf{h}_k^H \mathbf{G}\mathbf{w}_k|^2}{\sum_{j\neq k} |\mathbf{h}_k^H \mathbf{G}\mathbf{w}_j|^2 + \sigma_0^2}, \tag{24}$$

and the per-user SE is

$$R_k(\Phi^p, \mathbf{W}) = \log_2\big(1 + \mathrm{SINR}_k(\Phi^p, \mathbf{W})\big). \tag{25}$$

The learning objective of the LLM-based agent is to maximize an aggregate user utility based on the instantaneous spectral efficiencies. The per-step spectral efficiencies is written as

$$\mathrm{SE}_t = \sum_{k=1}^{K} R_k(\Phi^p_t, \mathbf{W}_t), \tag{26}$$

where $\mathrm{agent}_k(\cdot)$ is a per-user utility function.

4.1.2 Beamforming design

The beamforming latent produced by the actor, denoted $z_w \in \mathbb{R}^L$, is decoded by a small parametric PowerAllocator network into two sets of parameters, a per-user positive uplink factor vector $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_K]^{\mathsf{T}}$ and a normalized downlink power fraction vector $\mathbf{p} = [p_1, \ldots, p_K]^{\mathsf{T}}$ with $\sum_k p_k = 1$. In the implementation the mapping is

$$\boldsymbol{\lambda} = \mathrm{softplus}\big(f_\theta(z_w)\big) + \epsilon, \quad \mathbf{p} = \mathrm{softmax}\big(g_\theta(z_w)\big), \tag{27}$$

where f_θ and g_θ are the linear heads of the PowerAllocator network and $\epsilon > 0$ is a small constant for numerical stability. The PowerAllocator architecture and activation choices match the implementation in the environment.

Given the equivalent channel $\widetilde{\mathbf{H}} = \mathbf{HG} \in \mathbb{C}^{K\times NM}$ that already incorporates the pinching-phase matrix $\mathbf{G}$, the implemented precoder follows a regularized linear structure

$$\mathbf{W} \;=\; \widetilde{\mathbf{H}}^{\mathsf{H}}\Big(\Lambda\,\widetilde{\mathbf{H}}\,\widetilde{\mathbf{H}}^{\mathsf{H}} + \sigma^2\mathbf{I}_K\Big)^{-1}\mathbf{P}^{1/2}, \tag{28}$$

where $\Lambda = \mathrm{diag}(\boldsymbol{\lambda})$ and $\mathbf{P}^{1/2} = \mathrm{diag}(\sqrt{p_1}, \ldots, \sqrt{p_K})$. The inverse is implemented as a direct matrix inverse with fallback to a pseudo-inverse when necessary to avoid numerical singularities. Finally, $\mathbf{W}$ is normalized by its Frobenius norm to satisfy the total transmit-power constraint as

$$\mathbf{W}' = \frac{\mathbf{W}}{\|\mathbf{W}\|_F}. \tag{29}$$

With $\mathbf{W}$ and $\widetilde{\mathbf{H}}$ computed, we evaluate the SINR of each user using equation (24) , in the implementation the term $\mathbf{G}\mathbf{w}_k$ replaced by the corresponding columns of $\widetilde{\mathbf{H}}^{\mathsf{H}}\mathbf{W}$, and the resulting per-user rates determine the scalar reward defined in subsection 4.1.2.

4.1.3 MoDE-DRL for PASS optimization

To address the non-convex challenge outlined in section 2, we design a layered MoDE actor composed of two sequential stages. The first stage proposes pinching positions for the antennas, and the second produces beamforming latent codes. Below we detail the SAC state space, action space, and reward function:

- **State** s: The state is designed to include the environment and configuration variables relevant to antenna placement and beamforming. Concretely, the state contains user positions, current pinching positions, and the complex channel matrix represented by its real and imaginary parts. We therefore write the state space as
$$\mathcal{S} = \{\{\mathbf{k}_i\}, \{\mathbf{p}_i\}, \{r_i\}\}, \tag{30}$$
where user positions, stacked as $(x_1, y_1, \ldots, x_K, y_K)$, length $2K$; current pinching positions flattened over waveguides and antennas, length NM; the complex channel state $\mathbf{H}$ represented by its real and imaginary parts and flattened to length $2KMN$.
- **Action** a: The action space A is defined as the optimization process of pinching antenna position together with beamforming matrics W, which concatenates normalized initial per-waveguide positions, relative inter-antenna increments, and the beamforming latent vector z_w. The continuous action $a_t \in \mathbb{R}^A$ is partitioned as
$$a_t = \big[\mathbf{x}_1^{(n)},\ \Delta^{(n,m)},\ z_w\big],$$
where $\mathbf{x}_1^{(n)} \in \mathbb{R}^N$ contains normalized first-antenna positions per waveguide, in the LLM-based agent output range $[-1, 1]$ and mapped to physical coordinates by $x_{1,n}^p = (x_{1,n}^{(n)} + 1)\, D/2$; $\Delta^{(n,m)} \in \mathbb{R}^{N(M-1)}$ are normalized relative increments for subsequent antennas on each waveguide; each normalized delta is mapped to a feasible incremental distance that enforces the minimum spacing $\Delta_{\min}$ and ensures the last antenna remains within $[0, D]$; $z_w \in \mathbb{R}^L$ is the beamforming latent vector decoded by a small PowerAllocator network into per-user uplink factors $\{\lambda_k\} > 0$ and normalized downlink power fractions $\{p_k\}$ with $\sum_k p_k = 1$.
- **Reward** r: The reward function takes into account the objective function, that is, the sum SE R_k. It is designed as aggregating per-user utilities, and In the environment implementation used for training and evaluation, we take the per-episode step reward to be proportional to the sum SE with a small improvement bonus, namely
$$r_t = 10\Big(\mathrm{SE}_t + 0.1\big(\mathrm{SE}_t - \mathrm{SE}_{t-1}\big)\Big), \tag{31}$$
where SE_t is defined in equation (26) and SE_{t-1} denotes the previous step value; the increment term is omitted at the first step; the multiplicative factor is a numeric scaling chosen for training stability.

4.1.4 Experimental Results

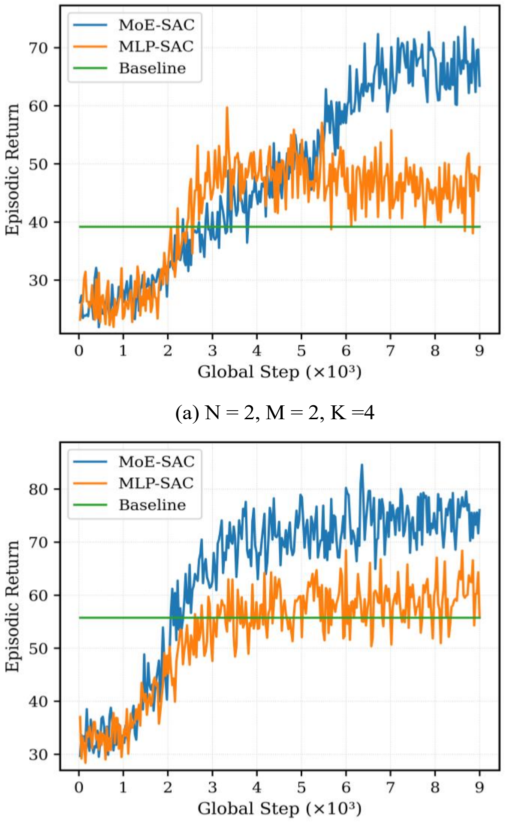
We consider a PASS case with N waveguides, each equipped with M pinching antennas. A total of K users are uniformly distributed in a $D \times D = 100 \times 100\,\mathrm{m}^2$ square region. The waveguide height is set to $d = 3\,\mathrm{m}$, the carrier frequency is $f_c = 28\,\mathrm{GHz}$, the effective refractive index is $n_{\text{eff}} = 1.4$, and minimum inter-antenna spacing is $\Delta_{\min} = \lambda_g$. Without loss of generality, We consider multiple training scenarios to demonstrate robustness across system sizes.

Fig. 4 plots the cumulative episodic return versus training iterations for MoDE with SAC, MLP with SAC and a deterministic baseline under identical random seeds and hyperparameters.. In the baseline, the first antenna on each waveguide is aligned with the nearest user in the x-direction, the remaining antennas are placed at the minimum spacing, and the transmit beamforming is implemented using zero-forcing (ZF). It shows the scenario where N = 2, M = 2 and K = 4 and the scenario using N = 7, M = 3 and K = 13. In both scenarios, the LLM-based agent of MoDE-SAC attains higher final episodic return than the LLM-based agent of MLP-SAC, and it surpasses the deterministic baseline after sufficient training.

Fig. 5 focuses on the effect of expert allocation across the two MoDE layers and displays the learning curves for these two allocations together with the deterministic baseline. It uses N = 3, M = 2 and K = 6. Here we fix the basic expert count to 3 and compare two layerwise allocations. Configuration A uses fine grained segmentation factor 1 for the position layer and factor 4 for the beamforming layer, which yields 3 experts for the position stage and 12 fine grained experts for the beamforming stage. Configuration B uses segmentation factor 2 for both layers, which yields 6 experts in each layer. The results show that allocating more experts to the beamforming layer produces higher converged episodic return than the symmetric allocation. The symmetric allocation still improves slightly upon the deterministic baseline, but it is outperformed by the beamforming heavy configuration.

Fig. 6 plots the late-training average episodic return as a function of number of experts for three methods including MoDE with SAC, MLP with SAC, and a heuristic baseline. The total expert count combines the base expert number with the fine-grained segmentation factors used in our MoDE actor. Results are reported for two environment configurations including a higher-complexity setting with $N = 5$, $M = 3$, $K = 10$ and a lower-complexity setting with $N = 3$, $M = 2$, $K = 6$.

In the higher-complexity environment MoDE improves as the expert count increases up to an intermediate optimum, after which performance degrades. This pattern



(a) N = 2, M = 2, K =4

(b) N = 7, M = 3, K =13

Fig. 4. Comparison of average return versus training steps for MoDE–SAC, MLP-SAC and the heuristic baseline in PASS.

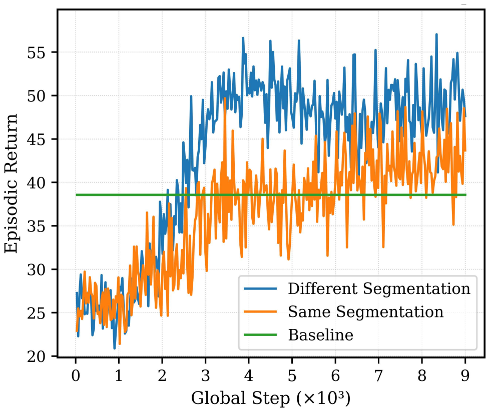
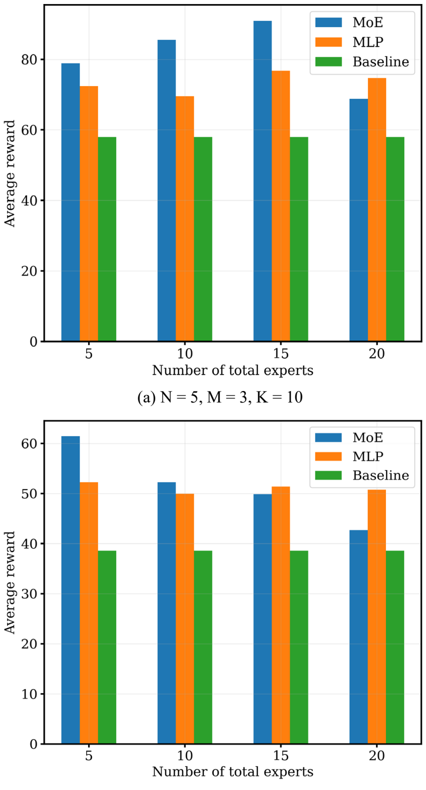


Fig. 5. Average return versus training steps for MoDE with different configuration of experts.

indicates that in a complex environment additional experts provide useful representational capacity and specialization up to a point. However, more experts also increase routing complexity and decreases the effective per-expert sample size , which amplifies optimization variance, leading to increased noise in parameter updates. Therefore experts can be underutilised or poorly trained , resulting harm performance. Conversely, in the lower-complexity environment, increasing the number of experts tends to decrease MoDE performance. This shows that when the task complexity is low, an overly large expert pool increases optimization difficulty and leads to worse late-stage performance.

Consequently there is a modality-dependent optimum. Moderate expert counts are preferred for complex environments while smaller expert pools are preferable for simpler environments.



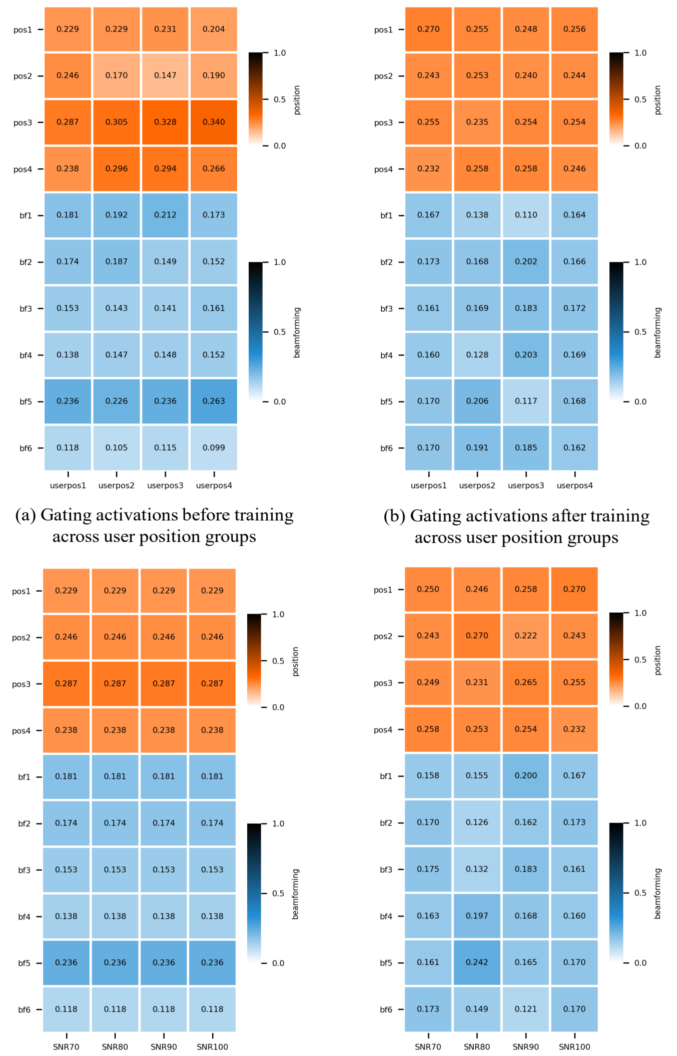
(a) N = 5, M = 3, K = 10

(b) N = 3, M = 2, K = 6

Fig. 6. Impact of total expert count on late-stage episodic return under different environment complexities.

Fig. 7 visualizes the gating activations produced by the MoDE actor under the PASS environment with $N = 3$, $M = 2$, and $K = 6$. We set a basic expert number count of 2, position segmentation of 2, and the beamforming sementation of 3. Each heatmap cell shows the router output probability for a single expert, taking values in $[0, 1]$. Rows 1 to 4 correspond to the position experts and rows 5 to 10 correspond to the beamforming experts . Columns 1 to 4 correspond to four environment groups, including variations such as different user-position configurations and different SNR settings.

Panels (a)/(c) show activations at an early stage, while panels (b)/(d) show activations at a later stage after training. Panels (a)/(b)shows the gating activation in different user position settings. Initially the routing strongly privileges ‘pos3’ for several user-position groups, reflecting an early bias toward a single positional expert. After training the distribution of position activations becomes more balanced for many groups, while beamforming experts develop distinct fingerprints across user sets. For instance, user-position set 2 after training shows relatively larger contributions from ‘bf5’ and ‘bf6’, while in user-positioin set 4 the gating activation is nearly equal, which demonstrates that beamforming experts have specialized to serve different spatial interference patterns produced by distinct user layouts.



(a) Gating activations before training across user position groups

(b) Gating activations after training across user position groups

(c) Gating activations before training across SNR groups

(d) Gating activations after training across SNR groups

Fig. 7. Gating activations across environment settings of user-position and SNR variations.

Panels (c)/(d) showcases the router activation in different SNR settings. At the early stage the router biases are largely stationary across SNR groups. The position expert 'pos3' has the largest share $\approx$ 0.287 and beamforming expert 'bf5' is also relatively large ≈ 0.236 in every column, indicating a weak dependence on SNR at this stage. After training, the router becomes more context-sensitive, that is, the position experts are more evenly utilized across SNR groups, while beamforming allocations vary with the SNR group. For example, 'bf5' is notably higher for SNR = 80 group, and 'bf1' shows larger contributions in SNR = 90 group. This change indicates that training induces specialization in the beamforming experts according to channel conditions, while position experts provide a relatively uniform positional encoding.

Taken together, the results demonstrate two main points. First, the routers learn to adapt expert selection to environmental context. Training shifts the router outputs from a near-single-expert bias toward a richer, instance-dependent allocation. Second, the MoDE structure yields functional specialization. Position experts encode coarse spatial decisions since they become more evenly engaged after training, while beamforming experts specialize to subtle differences between environment such as channel/SNR conditions and user-position sets. This specialization is precisely the property we intended to exploit, that is, by letting different experts focus on distinct sub-problems such as positioning and precoding, the actor can represent more complex, context-dependent policies without dramatically increasing per-step decision complexity.

We use these heatmaps as empirical evidence that the MoDE router achieves adaptive, interpretable gating. The router probabilities are direct, per-instance indicators of which experts the actor relies on, and their evolution from pre- to post-training illustrates the emergence of task-dependent expert specialization. It also shows that MoDE can (i) adapt to environment statistics such as SNR and user layout, and (ii) concentrate representational capacity where it is most needed during learning.

4.2 Joint decision-making problem of Power Allocation and Beamforming Design in MU-MIMO

4.2.1 MoDE-DRL for MU-MIMO Optimization

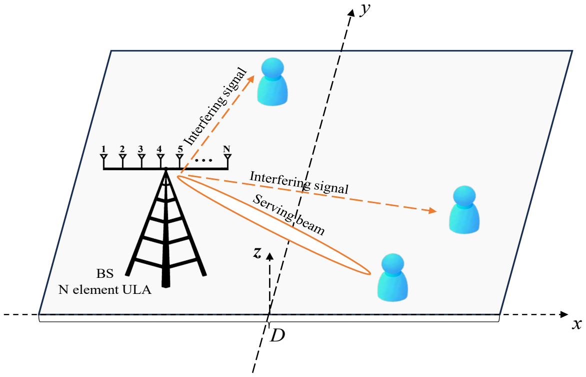


Fig. 8. MU-MIMO system model

We consider a downlink multi-user MIMO system as Fig. 8 where a BS equipped with N_t transmit antennas serves K single-antenna users. The instantaneous channel from the BS to user k is denoted $\mathbf{h}_k \in \mathbb{C}^{N_t}$, with the aggregate channel matrix $\mathbf{H} = [\mathbf{h}_1, \ldots, \mathbf{h}_K]$. The joint optimization of power allocation and beamforming parameters directly determines the achievable signal-to-interference-plus-noise ratios across users.

The objective is to maximize a network utility function defined over instantaneous user rates. Let $R_k(s, a)$ represent the SE of user k under state s and action a. We formulate the weighted sum-rate maximization problem as the equation (23).

To capture the distinct aspects of power allocation and beamforming design, the actor employs a layered mixture of experts. The first layer processes the observation through experts that output user power allocation factors of dimension K. The second layer concatenates the original observation with these power factors and routes the augmented input to beamforming experts that generate latent representations for precoding of dimension 24. This hierarchical structure ensures that power allocation decisions guide the subsequent beamforming optimization in an integrated manner.

In our SAC implementation, the state encompasses current channel measurements $\mathbf{H}$, user geographical positions, and antenna array geometry. The continuous action vector encodes both beamforming parameters and relaxed representations of discrete configuration choices, maintaining end-to-end differentiability. The reward function directly corresponds to the objective in equation (31), with entropy regularization in SAC encouraging exploration across diverse configuration-beamforming combinations and enhancing robustness in multimodal optimization landscapes.

4.2.2 Experimental Results

We evaluate the proposed LLM-based agent of MoDE-SAC on the joint power allocation and beamforming optimization task in MU-MIMO systems. The BS is equipped with N transmit antennas serving K single-antenna users uniformly distributed in a $D \times D$ square meter area. The system operates at a carrier frequency of $f_c = 28$ GHz with a path loss exponent of 2.5. We consider different training scenarios to demonstrate robustness across different system configurations.

Fig. 9 presents the average return versus training steps for MoDE with SAC, MLP with SAC , and a heuristic baseline under identical random seeds and hyperparameters. The baseline employs MMSE precoding combined with water-filling power allocation. The results show when $N = 7$, $K = 13$, $D = 50$ and when $N = 11$, $K = 18$, $D = 40$, the LLM-based agent of MoDE-SAC demonstrates faster convergence speed and achieves a higher final performance compared to the LLM-based agent of MLP-SAC, with both significantly outperforming the baseline after sufficient training.

4.3 Joint decision-making problem of IRS Phase and Beamforming Design in IRS-Aided MIMO

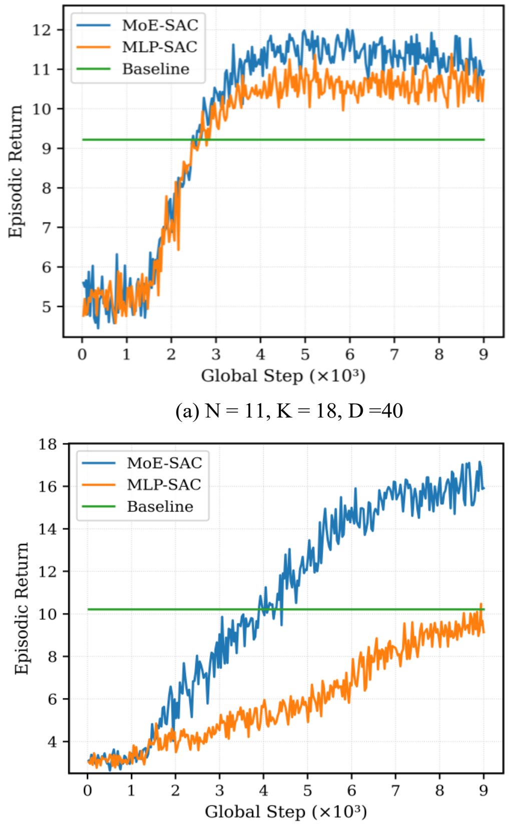
4.3.1 MoDE-DRL for IRS-aided-MIMO Optimization

We consider an IRS-assisted downlink MIMO system as Fig. 10 where a BS equipped with M transmit antennas serves K single-antenna users via both a direct link and an intelligent reflecting surface (IRS) with N passive reflecting elements. Let $\mathbf{G} \in \mathbb{C}^{N\times M}$ denote the BS to IRS channel, $\mathbf{d}_k \in \mathbb{C}^{M\times 1}$ the direct BS to user k channel, and $\mathbf{r}_k \in \mathbb{C}^{N\times 1}$ the IRS to user k channel. $\boldsymbol{\Phi} = \mathrm{diag}(e^{j\theta_1}, \dots, e^{j\theta_N})$ denotes the IRS diagonal phase-shift matrix, the effective channel from the BS to user k, which is a column vector of length M, is given by

$$\mathbf{h}_k^H = \mathbf{r}_k^H \boldsymbol{\Phi} \mathbf{G} + \mathbf{d}_k^H.$$

The joint optimization of IRS phase $\{\theta_1,\theta_2,\dots,\theta_n\}$ and the BS precoder $\mathbf{W} \in \mathbb{C}^{M\times K}$ determines the end-to-end SINR and hence the achievable spectral efficiencies.

The objective is to maximize the network sum-rate over instantaneous user rates. Let $R_k(s,a)$ represent the SE of user k under state s and action a. We formulate the weighted sum-rate maximization problem as the equation (23).



(a) N = 11, K = 18, D =40

(b) N = 7, K = 13, D =50

Fig. 9. Comparison of average return versus training steps for MoDE–SAC, MLP-SAC and the heuristic baseline in MU-MIMO.

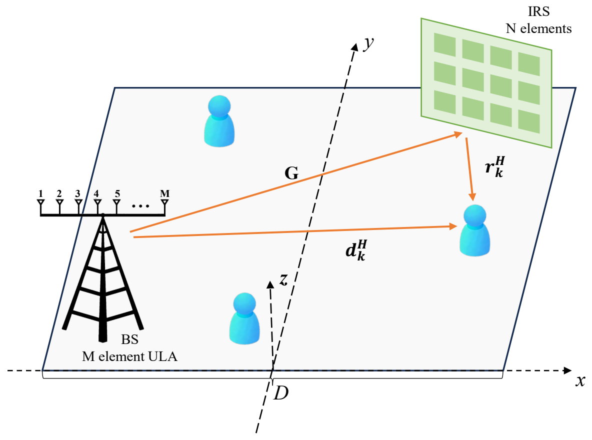


Fig. 10. IRS-aided-MIMO system model

To handle the mixed nature of IRS phase adjustments and continuous beamforming design, the actor employs a layered mixture of experts architecture. The first layer processes the environmental observation through specialized experts that generate IRS phase configurations, outputting both sine and cosine components of phase shifts with dimension $2N$. The second layer concatenates the original observation with these phase parameters and routes the augmented input to beamforming experts that produce latent representations for precoder optimization of dimension

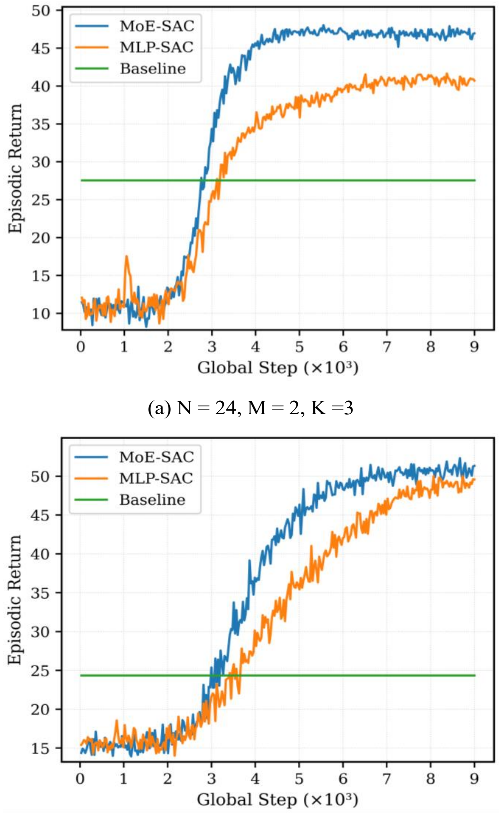
24. This hierarchical decomposition ensures coordinated optimization where IRS phase adjustments establish favorable propagation conditions that the beamforming module subsequently exploits.

In our SAC implementation, the state incorporates previous IRS phase shifts, current effective channel measurements $\mathbf{H}_k$, and user positions. The continuous action vector encodes both IRS phase parameters and beamforming latent variables, maintaining full differentiability throughout the optimization pipeline. The reward function directly corresponds to the objective in equation (31), with entropy regularization in SAC promoting exploration across diverse IRS-beamforming configurations and enhancing performance in complex multimodal optimization landscapes.

4.3.2 Experimental Results

We evaluate the proposed LLM-based agent of MoDE-SAC on the joint IRS phase and beamforming optimization task in IRS-aided downlink MISO systems. The system configuration includes an access point with M transmit antennas, an intelligent reflecting surface with N passive elements, and K single-antenna users uniformly distributed in a $30 \times 30\,\mathrm{m}^2$ square region. The system operates at a carrier frequency of $f_c = 3.5$ GHz with path loss exponents of 3.0 for direct links and 2 for IRS-reflected paths. The reference path loss is set to $P_{lo} = -20$ dB at $d_o = 1$ m, with an additional IRS gain of 10 dB enhancing the reflected signals.Two environment configurations are considered to demonstrate robustness across scales.

Fig. 11 present the average return versus training steps for MoDE with SAC, MLP with SAC, and an alternating optimization baseline under identical random seeds and hyperparameters. The baseline operates on local copies of environment channels and alternately updates the MMSE precoder and per-element IRS phases via closed-form phase updates. It shows the scenarios with $N = 24$, $M = 2$, $K = 3$, and with $N = 36$, $M = 4$, and $K = 6$. The results show that the LLM-based agent of MoDE-SAC demonstrates faster convergence speed and achieves higher final performance compared to the LLM-based agent of MLP-SAC. In both configurations, the proposed LLM-based agents of MoDE-SAC and MLP-SAC significantly outperform the alternating optimization baseline after sufficient training.



(a) N = 24, M = 2, K =3

(b) N = 36, M = 4, K =6

Fig. 11. Comparison of average return versus training steps for MoDE–SAC, MLP-SAC and the heuristic baseline in IRS-aided-MIMO.

5 Conclusion

We proposed a novel approach to enhancing user SE in next-generation wireless networks, focusing on joint physical-layer configuration and resource allocation. Our primary contribution is the MoDE-DRL algorithm, designed to address the tightly coupled optimization challenges in high-dimensional, non-convex decision spaces. By employing an MoDE-aided LLM-based agent that supplies real-time, sample-efficient feedback on SE, the framework enables coordinated adaptation of configuration and signal; the results presented in this work reflect the joint optimizations carried out across the three representative scenarios studied, PASS, MU-MIMO,and IRS-aided-MIMO. Moreover, the proposed architecture naturally extends to other coupled wireless problems where the layered MoDE assigns specialized experts to complementary subtasks. Overall, our methodology demonstrates the potential of combining MoDE architectures with DRL to enhance adaptive resource allocation, paving the way for real-time resource management in large-scale 6G wireless networks.

References

[1] H. F. Alhashimi, M. N. Hindia, K. Dimyati, E. B. Hanafi, N. Safie, F. Qamar, K. Azrin, and Q. N. Nguyen, “A survey on resource management for 6g heterogeneous networks: current research, future trends, and challenges,” *Electronics*, vol. 12, no. 3, p. 647, 2023.

[2] W. Saad, M. Bennis, and M. Chen, “A vision of 6g wireless systems: Applications, trends, technologies, and open research problems,” *IEEE network*, vol. 34, no. 3, pp. 134–142, 2019.

[3] S. Andreev, V. Petrov, M. Dohler, and H. Yanikomeroglu, “Future of ultra-dense networks beyond 5g: Harnessing heterogeneous moving cells,” *IEEE Communications Magazine*, vol. 57, no. 6, pp. 86–92, 2019.

[4] R. J. Mailloux, *Phased array antenna handbook*. Artech house, 2017.

[5] Q. Wu and R. Zhang, “Intelligent reflecting surface enhanced wireless network via joint active and passive beamforming,” *IEEE transactions on wireless communications*, vol. 18, no. 11, pp. 5394–5409, 2019.

[6] Y. Zeng, R. Zhang, and T. J. Lim, “Wireless communications with unmanned aerial vehicles: Opportunities and challenges,” *IEEE Communications magazine*, vol. 54, no. 5, pp. 36–42, 2016.

[7] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, “A survey on mobile edge computing: The communication perspective,” *IEEE communications surveys & tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.

[8] F. Guo, F. R. Yu, H. Zhang, X. Li, H. Ji, and V. C. Leung, "Enabling massive iot toward 6g: A comprehensive survey," *IEEE Internet of Things Journal*, vol. 8, no. 15, pp. 11 891–11 915, 2021.

[9] M. Singh and G. Baranwal, "Quality of service (qos) in internet of things," in *2018 3rd International Conference On Internet of Things: Smart Innovation and Usages (IoT-SIU)*. IEEE, 2018, pp. 1–6.

[10] Y. Chen, S. Zhang, S. Xu, and G. Y. Li, "Fundamental trade-offs on green wireless networks," *IEEE Communications Magazine*, vol. 49, no. 6, pp. 30–37, 2011.

[11] Y.-F. Liu, T.-H. Chang, M. Hong, Z. Wu, A. M.-C. So, E. A. Jorswieck, and W. Yu, "A survey of recent advances in optimization methods for wireless communications," *IEEE Journal on Selected Areas in Communications*, 2024.

[12] K. B. Letaief, W. Chen, Y. Shi, J. Zhang, and Y.-J. A. Zhang, "The roadmap to 6g: Ai empowered wireless networks," *IEEE communications magazine*, vol. 57, no. 8, pp. 84–90, 2019.

[13] Z. Feng, Z. Wei, X. Chen, H. Yang, Q. Zhang, and P. Zhang, "Joint communication, sensing, and computation enabled 6g intelligent machine system," *IEEE Network*, vol. 35, no. 6, pp. 34–42, 2022.

[14] Y. Shi, L. Lian, Y. Shi, Z. Wang, Y. Zhou, L. Fu, L. Bai, J. Zhang, and W. Zhang, "Machine learning for large-scale optimization in 6g wireless networks," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 4, pp. 2088–2132, 2023.

[15] X. Xie, F. Fang, and Z. Ding, "Joint optimization of beamforming, phase-shifting and power allocation in a multi-cluster irs-noma network," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 8, pp. 7705–7717, 2021.

[16] M. S. Elbamby, C. Perfecto, C.-F. Liu, J. Park, S. Samarakoon, X. Chen, and M. Bennis, "Wireless edge computing with latency and reliability guarantees," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1717–1737, 2019.

[17] E. Bastug, M. Bennis, M. Médard, and M. Debbah, "Toward interconnected virtual reality: Opportunities, challenges, and enablers," *IEEE Communications Magazine*, vol. 55, no. 6, pp. 110–117, 2017.

[18] Y. Lu, Z. Zhang, and L. Dai, "Hierarchical beam training for extremely large-scale mimo: From far-field to near-field," *IEEE Transactions on Communications*, vol. 72, no. 4, pp. 2247–2259, 2023.

[19] K. Shen, W. Yu, L. Zhao, and D. P. Palomar, "Optimization of mimo device-to-device networks via matrix fractional programming: A minorization–maximization approach," *IEEE/ACM Transactions on Networking*, vol. 27, no. 5, pp. 2164–2177, 2019.

[20] K. Shen and W. Yu, "Fractional programming for communication systems—part i: Power control and beamforming," *IEEE Transactions on Signal Processing*, vol. 66, no. 10, pp. 2616–2630, 2018.

[21] N. C. Luong, D. T. Hoang, S. Gong, D. Niyato, P. Wang, Y.-C. Liang, and D. I. Kim, "Applications of deep reinforcement learning in communications and networking: A survey," *IEEE communications surveys & tutorials*, vol. 21, no. 4, pp. 3133–3174, 2019.

[22] W. Cai, J. Jiang, F. Wang, J. Tang, S. Kim, and J. Huang, "A survey on mixture of experts in large language models," *IEEE Transactions on Knowledge and Data Engineering*, 2025.

[23] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean, "Outrageously large neural networks: The sparsely-gated mixture-of-experts layer," *arXiv preprint arXiv:1701.06538*, 2017.

[24] B. Pan, Y. Shen, H. Liu, M. Mishra, G. Zhang, A. Oliva, C. Raffel, and R. Panda, "Dense training, sparse inference: Rethinking training of mixture-of-experts language models," *arXiv preprint arXiv:2404.05567*, 2024.

[25] X. Nie, X. Miao, S. Cao, L. Ma, Q. Liu, J. Xue, Y. Miao, Y. Liu, Z. Yang, and B. Cui, "Evomoe: An evolutional mixture-of-experts training framework via dense-to-sparse gate," *arXiv preprint arXiv:2112.14397*, 2021.

[26] A. Bereyhi, S. Asaad, C. Ouyang, Z. Ding, and H. V. Poor, "Downlink beamforming with pinching-antenna assisted mimo systems," *arXiv preprint arXiv:2502.01590*, 2025.

[27] Z. Ding, R. Schober, and H. V. Poor, "Flexible-antenna systems: A pinching-antenna perspective," *IEEE Transactions on Communications*, 2025.

[28] X. Xu, X. Mu, Z. Wang, Y. Liu, and A. Nallanathan, "Pinching-antenna systems (pass): Power radiation model and optimal beamforming design," *arXiv preprint arXiv:2505.00218*, 2025.

[29] Q. Shi, M. Razaviyayn, Z.-Q. Luo, and C. He, "An iteratively weighted mmse approach to distributed sum-utility maximization for a mimo interfering broadcast channel," *IEEE Transactions on Signal Processing*, vol. 59, no. 9, pp. 4331–4340, 2011.

[30] T. Yoo and A. Goldsmith, "On the optimality of multiantenna broadcast scheduling using zero-forcing beamforming," *IEEE Journal on selected areas in communications*, vol. 24, no. 3, pp. 528–541, 2006.

[31] C. Li, X. Wang, L. Yang, and W.-P. Zhu, "A joint source and relay power allocation scheme for a class of mimo relay systems," *IEEE Transactions on Signal Processing*, vol. 57, no. 12, pp. 4852–4860, 2009.

[32] E. Ali, M. Ismail, R. Nordin, and N. F. Abdulah, "Beamforming techniques for massive mimo systems in 5g: overview, classification, and trends for future research," *Frontiers of Information Technology & Electronic Engineering*, vol. 18, no. 6, pp. 753–772, 2017.

[33] F. W. Vook, A. Ghosh, and T. A. Thomas, "Mimo and beamforming solutions for 5g technology," in *2014 IEEE MTT-S International Microwave Symposium (IMS2014)*. IEEE, 2014, pp. 1–4.

[34] P. Wang, J. Fang, X. Yuan, Z. Chen, and H. Li, "Intelligent reflecting surface-assisted millimeter wave communications: Joint active and passive precoding design," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 14 960–14 973, 2020.

[35] H. Xie, J. Xu, and Y.-F. Liu, "Max-min fairness in irs-aided multicell miso systems with joint transmit and reflective beamforming," *IEEE Transactions on Wireless Communications*, vol. 20, no. 2, pp. 1379–1393, 2020.

[36] X. Li, J. Fang, F. Gao, and H. Li, "Joint active and passive beamforming for intelligent reflecting surface-assisted massive mimo systems," *arXiv preprint arXiv:1912.00728*, 2019.

[37] C. Huang, A. Zappone, G. C. Alexandropoulos, M. Debbah, and C. Yuen, "Reconfigurable intelligent surfaces for energy efficiency in wireless communication," *IEEE transactions on wireless communications*, vol. 18, no. 8, pp. 4157–4170, 2019.

[38] W. Fedus, B. Zoph, and N. Shazeer, "Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity," *Journal of Machine Learning Research*, vol. 23, no. 120, pp. 1–39, 2022.

[39] D. Dai, C. Deng, C. Zhao, R. Xu, H. Gao, D. Chen, J. Li, W. Zeng, X. Yu, Y. Wu *et al.*, "Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models," *arXiv preprint arXiv:2401.06066*, 2024.

[40] M. Haklidir and H. Temeltaş, "Guided soft actor critic: A guided deep reinforcement learning approach for partially observable markov decision processes," *IEEE Access*, vol. 9, pp. 159 672–159 683, 2021.

[41] Q. Wang, K. Feng, X. Li, and S. Jin, "Precodernet: Hybrid beamforming for millimeter wave systems with deep reinforcement learning," *IEEE Wireless Communications Letters*, vol. 9, no. 10, pp. 1677–1681, 2020.