MoDE: Mixture-of-Decision-Experts for DRL in High-Dimensional Coupled Wireless Optimization

Shiyi Lin, Hongyang Du

***Abstract*—Next-generation wireless networks require ultra reliable connectivity, extreme data rates and low latency for applications such as immersive communications and intelligent transportation. Achieving these targets in real time is challenging because design optimization problems are usually tightly coupled, high-dimensional and nonconvex. Existing approaches, whether classical model-based or generic learning-based methods, struggle to balance computational efficiency, stability, and scalability in such settings. To address this challenge and enhance system performance, we propose the Mixture-of-Decision-Experts (MoDE) integrated with Deep Reinforcement Learning (DRL), which effectively decomposes complex wireless control tasks into heterogeneous expert modules with adaptive routing, reducing cross-task interference and improving specialization. Evaluated on representative wireless tasks, MoDE-aided DRL outperforms a conventional Multilayer Perceptron (MLP)-based DRL baseline and a low-complexity heuristic baseline; furthermore, allocating more experts to higher-dimensional decision layers yields additional gains over uniform-expert-allocations design, demonstrating both efficiency and robustness.**

***Index Terms*—Generative AI (GAI), Wireless Networks, DRL, Mixture of Experts (MoE)**

I. Introduction

Recently, the relentless growth of wireless applications, from immersive extended-reality services to intelligent transportation, has propelled evolution toward Sixth-Generation (6G) networks. These systems aim to deliver ubiquitous connectivity, extreme data rates, and low-latency services [1], whose deployments are becoming increasingly heterogeneous and ultra-dense, incorporating new components such as Reconfigurable Intelligent Surfaces (RIS), Unmanned Aerial Vehicles (UAVs), and edge nodes [2]. While this architectural shift unlocks diversified Quality-of-Service (QoS), superior performance metrics like enhanced Spectral Efficiency (SE) [3], the same innovations also transform design tasks into large-scale, tightly coupled, nonconvex optimization problems that must be solved under strict real-time constraints [4].

Prior work has pursued model-based and learning-based approaches. For near-field beam training, the authors in [5] propose a multi-resolution hierarchical beam-training framework that designs polar-domain codebooks and applies a coarse-to-fine search to reduce overhead in antenna positioning and beam alignment; however, the scheme depends on precomputation, iterative solves and careful initialization, which limits adaptability and real-time use in large, dynamic, heterogeneous systems; As for learning-based approaches, Deep Reinforcement Learning (DRL) avoid explicit modeling by learning policies from interaction and therefore have been applied to many nonconvex and coupled optimization in wireless network such as beam management and UAVs control [6], but they often suffer from sample inefficiency, instability, and difficulty scaling to very large action spaces. These limitations motivate the following research question:

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

Architectures that decompose joint decisions into specialized subproblems are a natural remedy [7], thereby increasing representational capacity in a controlled way while maintaining stable learning throughout continuous multimodal control. Mixture-of-experts (MoE) models distribute capacity across specialized experts via learned routing [8]; recent work in Large Language Models also show that dense-activation variants help stabilize expert initialization and encourage richer specialization before sparse routing [9]. Motivated by these ideas, we propose MoDE-aided DRL framework tailored for tightly-coupled wireless optimization problems. Specifically, MoDE has three design features: (i) ***Layered task decomposition.*** Each layer produces the layer-specific action component, reducing dimensionality and clarifying learning targets. (ii) ***Expert specialization*** [10]. MoDE allows heterogeneous expert counts and internal segmentation, where high-complexity layers receive more experts, which improves representational fidelity and stabilizes learning in composite action spaces. (iii) ***Adaptive routing [11] with dense activation***. MoDE dynamically selects and combines experts based on state, which intelligently allocates computation to the most relevant experts, promoting specialization and stable training. Algorithmically, MoDE is unified and agnostic to deployment; it can run on a single device or be partitioned across platforms for distributed execution.

To demonstrate the practical relevance and generality of MoDE, we evaluate it on three representative classes of tightly-coupled wireless control problems. Although these scenarios rely on different physical mechanisms, they share a common multivariable, tightly-coupled structure that produces high-dimensional, nonconvex joint objectives, precisely the problem class MoDE targets. The main contributions of this paper are summarized as follows.

- We propose MoDE, a general-purpose MoE that can be readily integrated into various DRL frameworks for structured decision-making. MoDE factorizes tightly-coupled, high-dimensional, and 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 algorithmic instantiations for Pinching Antenna Systems (PASS), Multi-User Multiple-Input Multiple-Output (MU–MIMO), and RIS-aided MIMO cases.
- We perform many experiments showing that MoDE improves convergence and final utility relative to an MLP-

based DRL baseline and a low-complexity heuristic baseline; we also demonstrate that allocating more experts to higher-dimensional layers yields additional gains.

II. Proposed MoDE-aided DRL

A. MoDE-Based Actor Network Architecture

Our two-stage MoDE jointly predicts configuration action and high-dimensional signal action. As shown in Fig. 1, the MoDE-based actor network consists of:

1) Layered MoDE with Heterogeneous Expert Allocation

Applying a conventional MoE actordirectly to high-dimensional coupled wireless optimization poses two challenges: (i) ***Coupled learning difficulty.*** Configuration decisions and signal-level actions require different feature sensitivities and inductive biases, so training a single-layer MoE induces noisy gradient updates and routing conflicts. (ii) ***Slow convergence and exploration cost.*** Large, multi-stage action spaces force routers to learn for both tasks, which increases exploration cost and delays expert specialization. To address these, we decompose the decision process into two dedicated MoDE layers so each layer can specialize in its own subtask, maintaining computational efficiency and achieving higher specialization without interference from other tasks:

- Configuration MoDE layer: Routes the state to a small set of experts to produce a structural configuration action.
- Signal MoDE layer: Takes an enriched state $\tilde{s} = [s; a_{\text{config}}]$, where a_{config} denotes the configuration action. Then we rout the enriched state $\tilde{s}$ to a larger set of experts for final signal-level optimization.

This layered design allows each stage to specialize independently. We allocate more experts to the higher-dimensional and nonlinear-mapping signal layer to increase representational capacity where it is most needed, while using fewer experts for the simpler configuration task to avoid unnecessary overparameterization and reduce routing overhead.

2) Fine-Grained Expert Segmentation and Shared Experts Isolation

In conventional MoE, limited experts model diverse, sometimes incompatible patterns, reducing specialization and affecting overall performance [12]. We therefore introduce ***fine-grained segmentation***: each expert is split into smaller sub-experts with reduced hidden dimensions that are activated together, allowing finer representation learning [10]. To prevent redundancy from overlapping common knowledge, we also introduce a ***dedicated shared expert*** in each layer that captures general features, freeing the segmented experts to focus on specialized patterns [13]. Together, segmentation enhances specialization, and isolation centralize generalization, creating a complementary mechanism. Both experts in MoDE are implemented as two-hidden-layer networks that output mean μ and log-standard-deviation $\log\sigma$ of a Gaussian policy. The Expert Feed-Forward Network (ExpertFFN) computation are:

$$\begin{cases} h_1 = \text{ReLU}(W_1 x + b_1), \\ h_2 = \text{ReLU}(W_2 h_1 + b_2), \\ \mu = W_\mu h_2 + b_\mu, \\ log\sigma = W_\sigma h_2 + b_\sigma, \end{cases} \tag{1}$$

where x is the input vector, 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. Hence, Fine-Grained Experts outputs are

$$\begin{cases} (\mu_i^{\text{config}}, \log\sigma_i^{\text{config}}) = E_i^{\text{config}}(s), & i = 1 \dots n, \\ (\mu_j^{\text{signal}}, \log\sigma_j^{\text{signal}}) = E_j^{\text{signal}}(\tilde{s}), & j = 1 \dots m, \end{cases} \tag{2}$$

where n denote the segmentation factors for the configuration experts and m for signal experts. Each MoDE layer employs a lightweight router computing normalized gating weights $\mathbf{w} = \frac{\exp((R\mathbf{x}+\mathbf{r})_i)}{\sum_{j=1}^{E}\exp((R\mathbf{x}+\mathbf{r})_j)}$, where x is the layer input, R and $\mathbf{r}$ are router parameters, E is the number of fine-grained experts in that layer. Using the per-sample weights w_i, the mixed distribution parameters for configuration layer are computed by adding the shared expert's output with a weighted sum of fine experts:

$$\begin{cases} \mu^{\text{config}}(s) = \mu_{\text{shared}}^{\text{config}}(s) + \sum_{i=1}^{n_{\text{fine}}} w_i^{\text{config}}(s)\mu_i^{\text{config}}(s), \\ \log\sigma^{\text{config}}(s) = \log\sigma_{\text{shared}}^{\text{config}}(s) + \sum_{i=1}^{n_{\text{fine}}} w_i^{\text{config}}(s)\log\sigma_i^{\text{config}}(s). \end{cases} \tag{3}$$

We then sample the intermediate configuration action as $a_{\textbf{config}} = \tanh\left(\mu^{\text{config}}(s) + \sigma^{\text{config}}(s)\cdot\epsilon\right), \quad \epsilon \sim \mathcal{N}(0, I)$, and similarly for the signal layer using enriched state $\tilde{s}$. The final action vector a is the concatenation of configuration and signal actions as $a = \text{scale}\cdot[a_{\textbf{config}}\ ;\ a_{\textbf{signal}}] + \text{bias}$.

3) Load Balance Consideration

Automatically learned routers may suffer from routing collapse by few experts monopolizing most samples and persistent load imbalance by unequal expert utilization, both of which reduce effective capacity and slow convergence [12]. We therefore add two lightweight regularizers as follows. A ***diversity loss*** encourages high entropy in the per-sample routing distribution to mitigate routing collapse:

$$\begin{cases} \bar{H}_{\text{config}} = \frac{1}{B}\sum_{b=1}^{B}\left(-\sum_{i=1}^{E_{\text{config}}} w_{b,i}^{\text{config}}\log w_{b,i}^{\text{config}}\right), \\ \bar{H}_{\text{signal}} = \frac{1}{B}\sum_{b=1}^{B}\left(-\sum_{i=1}^{E_{\text{signal}}} w_{b,i}^{\text{signal}}\log w_{b,i}^{\text{signal}}\right), \\ \mathcal{L}_{\text{diversity}} = -\left(\bar{H}_{\text{config}} + \bar{H}_{\text{signal}}\right), \end{cases} \tag{4}$$

where $\bar{H}$ represents mean entropy, B is batch size, E is the number of experts, and $w_{b,i}$ is normalized routing weight of each layer in MoDE for sample b and expert i. This term is minimized when the router assigns weights more evenly within each sample, reducing the chance of collapse to one expert. A

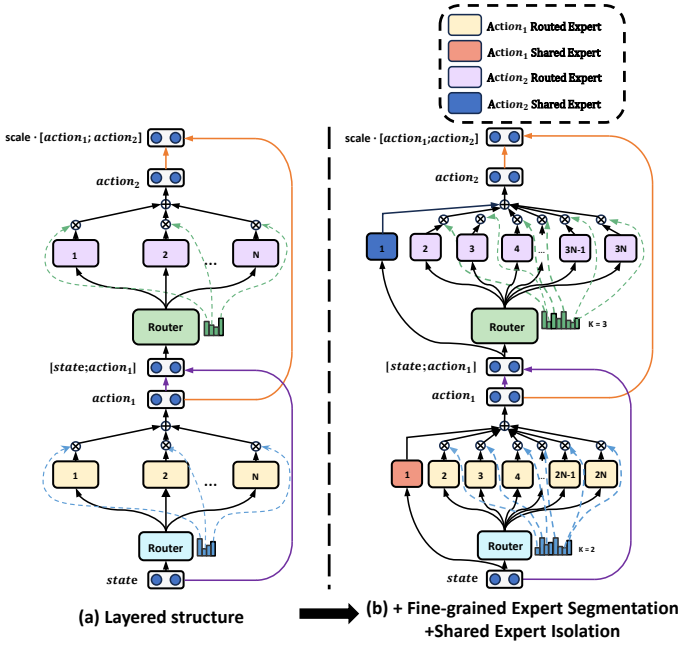


Fig. 1: Illustration of general MoDE. Subfigure (a) shows the layered MoDE structure. Subfigure (b) shows the fine-grained expert segmentation and the shared expert isolation strategy.

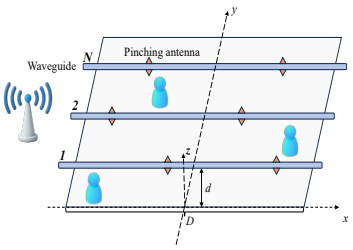
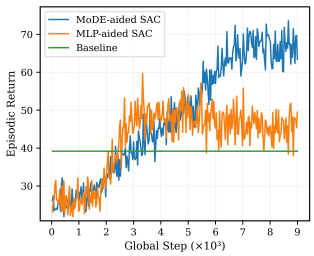
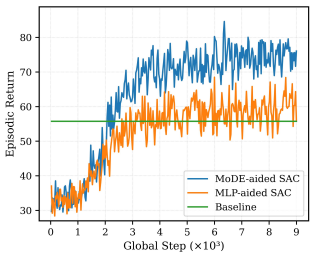


Fig. 2: PASS system model



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



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

Fig. 3: Average return versus training steps in PASS.

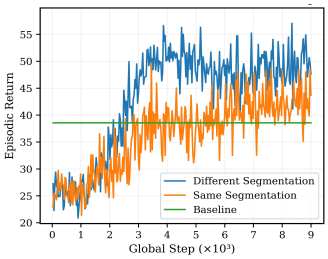


Fig. 4: MoDE with different experts-configuration.

Balance loss promotes equal expert utilization by regulating the long-term average usage of experts across a batch:

$$\begin{cases} \mu_i^{\text{config}} = \frac{1}{B}\sum_{b=1}^{B} w_{b,i}^{\text{config}}, \\ \mu_i^{\text{signal}} = \frac{1}{B}\sum_{b=1}^{B} w_{b,i}^{\text{signal}}, \\ \mathcal{L}_{\text{balance}} = \frac{1}{E}\sum_{i=1}^{E_{\text{config}}} \left(\mu_i^{\text{config}}\right)^2 + \frac{1}{E}\sum_{i=1}^{E_{\text{signal}}} \left(\mu_i^{\text{signal}}\right)^2. \end{cases} \tag{5}$$

This loss reaches its minimum when all experts are equally utilized on average, i.e., $\mu_i = \frac{1}{E}$. The total expert loss is $\mathcal{L}_{\text{expert}} = \mathcal{L}_{\text{diversity}} + \lambda \mathcal{L}_{\text{balance}}$, with $\lambda = 10^{-3}$ in our experiments to prevent collapse while encouraging global balance.

B. DRL Feedback Framework

MoDE-based actor network can be combined with a variety of DRL algorithms that support continuous control by providing structured, modular action generation and routing. In our experiment, we use SAC as a representative off-policy, entropy-regularized algorithm. The MoDE-based actor network yields the policy $\pi_\theta(s)$, while two critics network Q_ϕ, Q_ψ estimate state-action values independently to mitigate overestimation bias. Training proceeds over E episodes. At each step, the actor samples an action a from $\pi_\theta(s)$, and the environment returns a user-centric reward r, typically formulated based on utility metrics. These transitions (s, a, r, s') are stored in the replay buffer and later sampled to update the actor and critics via gradient descent. Through iterative training across episodes, the policy progressively refines its decision-making capacity, converging toward a robust and effective controller for complex wireless optimization tasks.

C. Overall Complexity

We analyze the parameter complexity of the proposed MoDE-based actor relative to a standard two-layer MLP baseline. Let O be the state dimension, P and B the configuration and signal action dimensions, and action $A = P+B$. For MoDE, let H be the shared hidden dimension of an expert, a the base number of experts, and b, c the segmentation factors for signal and configuration layers respectively. This yields configuration experts $K_p = a \cdot c$ and signal experts $K_b = a \cdot b$, each with a reduced hidden dimension of $H/2$. As each ExpertFFN consists of two dense internal linear layers plus two output heads, with input I, hidden width H and output O, the parameter count is $\mathrm{P}_{\text{expert}}(I, H, O) = (I \cdot H + H) + (H \cdot H + H) + (H \cdot O + O) + (H \cdot O + O)$. Using this building block, the total parameters for MoDE are the sum of its components, $\mathrm{P}_{\text{MoDE}} = K_p \cdot \mathrm{P}_{\text{expert}}(O, H_{e,p}, P) + K_b \cdot \mathrm{P}_{\text{expert}}(O+P, H_{e,b}, B) + \mathrm{P}_{\text{shared\_pos}}(O, H, P) + \mathrm{P}_{\text{shared\_bf}}(O+P, H, B) + \mathrm{P}_{\text{routers}}$, where each router is a simple linear layer. The dominant terms are quadratic in H expressed as $\mathrm{P}_{\text{MoDE}} = O\left(a\left(OH + H(P+B) + H^2\left(\frac{1}{b} + \frac{1}{c}\right)\right) + H^2\right)$. For the two-layer MLP actor with hidden width H_{MLP} plus two output heads, we have $\mathrm{P}_{\text{MLP}} = O(H_{\text{MLP}}^2 + H_{\text{MLP}}(O+A))$. Retaining only the leading quadratic terms, the parameter ratio approximates as $\frac{\mathrm{P}_{\text{MoDE}}}{\mathrm{P}_{\text{MLP}}} \approx \frac{C_{\text{MoDE}}\, H^2}{H_{\text{MLP}}^2}$, with $C_{\text{MoDE}} = a\left(\frac{1}{b} + \frac{1}{c}\right) + O\left(\frac{O}{H}, \frac{A}{H}\right)$. Thus, the relative overhead depends primarily on (i) the base expert count a, the segmentation factors b, c and (ii) the choice of the MoDE shared hidden width H and the MLP hidden width H_{MLP}. In practice, MoDE intentionally increase representational capacity in a controlled way, enhancing representational power without compromising convergence.

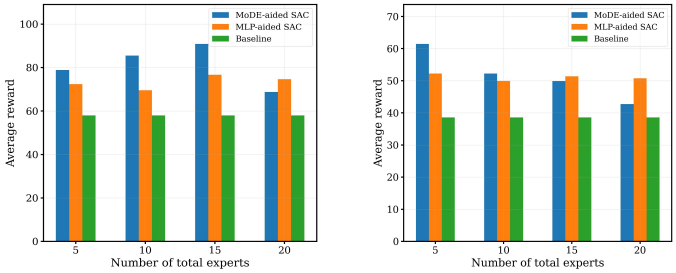
III. Cases

In this section, we detail and evaluate the MoDE-aided SAC framework on three tightly coupled, high-dimensional wireless optimization: PASS, MU-MIMO, and RIS-aided MIMO.

A. Joint Pinching ositions and Beamforming in PASS

1) MoDE-Aided SAC In PASS System Model

We consider a downlink multi-user PASS as Fig. 2 with N waveguides at Base Station (BS), M pinching antennas per waveguide and K users deployed in a $100 \times 100\,\text{m}^2$ region. The waveguide height is $d = 3\,\text{m}$, and the carrier frequency is $f_c = 28\,\text{GHz}$. For an antenna at location $\psi_{m,n}^p$ and user at ψ_k, the channel is $h_{n,m,k} = \frac{\sqrt{\eta}\,\exp\left(-j\frac{2\pi}{\lambda}\|\psi_k - \psi_{m,n}^p\|\right)}{\|\psi_k - \psi_{m,n}^p\|}$, where $\lambda = \frac{c}{f_c}$ is the free-space wavelength, and $\eta = \frac{c}{2\pi f_c}$ is the propagation constant. The per-waveguide feed-induced phases are aggregated into $\mathbf{G} = \text{blockdiag}(\mathbf{g}_1, \ldots, \mathbf{g}_N)$. With precoder $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_K]$ and transmitted symbols x_k, the received signal is $y_k = \mathbf{h}_k^H \mathbf{G}\,\mathbf{w}_k\, x_k + \sum_{j \neq k} \mathbf{h}_k^H \mathbf{G}\,\mathbf{w}_j\, x_j + n_k$, where $n_k \sim \mathcal{CN}(0, \sigma_0^2)$ is Gaussian noise, yielding $\text{SINR}_k(\Phi^p, \mathbf{W}) = \frac{|\mathbf{h}_k^H \mathbf{G}\mathbf{w}_k|^2}{\sum_{j \neq k} |\mathbf{h}_k^H \mathbf{G}\mathbf{w}_j|^2 + \sigma_0^2}$. Our objective is to maximize the sum SE defined as $\text{SE}_t = \sum_{k=1}^{K} \log_2(1 + \text{SINR}_k)$ across steps. The beamforming latent z_w is decoded into a per-user positive uplink factor vector $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_K]^\mathsf{T}$ and a normalized downlink power fraction vector $\mathbf{p} = [p_1, \ldots, p_K]^\mathsf{T}$ via lightweight neural network heads, ensuring $\lambda_k > 0$ and $\sum_k p_k = 1$. Given the equivalent channel $\widetilde{\mathbf{H}} = \mathbf{H}\mathbf{G}$, we use a practical regularized linear precoder $\mathbf{W} = \widetilde{\mathbf{H}}^\mathsf{H}\left(\Lambda \widetilde{\mathbf{H}}\widetilde{\mathbf{H}}^\mathsf{H} + \sigma^2 \mathbf{I}_K\right)^{-1} \mathbf{P}^{1/2}$, with $\Lambda = \text{diag}(\boldsymbol{\lambda})$ and $\mathbf{P}^{1/2} = \text{diag}(\sqrt{p_1}, \ldots, \sqrt{p_K})$. Finally $\mathbf{W}$ is normalized to satisfy the total transmit-power constraint as $\mathbf{W}' = \frac{\mathbf{W}}{\|\mathbf{W}\|_F}$. In MoDE, the configuration layer outputs positions of pinching-antenna and the signal layer produces beamforming latent vectors z_w. As for SAC, the state s contains user coordinates, current antenna positions, and the complex channel $\mathbf{H}$ represented by real/imag parts. The continuous action a concatenates normalized first-antenna positions per waveguide, normalized relative increments for the remaining antennas Δ, and the beamforming latent z_w. The reward is proportional to sum SE as $r_t = 10\Big(\text{SE}_t + 0.1\big(\text{SE}_t - \text{SE}_{t-1}\big)\Big)$,



(a) N = 5, M = 3, K = 10 (b) N = 3, M = 2, K = 6

Fig. 5: Impact of total expert count on late-stage episodic return under different environment complexities.

where SE_{t-1} denotes the previous step value; the multiplicative factor is a numeric scaling chosen for training stability.

2) Experimental Results

We compare the MoDE-aided SAC agent against an MLP-aided SAC baseline and a heuristic baseline with nearest-user antenna placement and Zero-Forcing precoding. Fig. 3 plots the average episodic return versus training iterations under two environment configurations: (a) $N = 2, M = 2, K = 4$ and (b) $N = 7, M = 3, K = 13$. MoDE-aided SAC attains higher final return than MLP-aided SAC , with both significantly outperforming the heuristic baseline after sufficient training.

Fig. 4 investigates the impact of expert allocation between the position and beamforming layers under $N = 3, M = 2, K = 6$ setting. Here we fix the basic expert count to 3 and compare two allocations: Configuration A uses segmentation factor 1 for position layer and factor 4 for beamforming layer; Configuration B uses segmentation factor 2 for both layers. The results show that allocating more experts to the beamforming layer produces higher converged episodic return than the symmetric allocation, which itself improves mildly over the heuristic baseline.

Fig. 5 summarizes late-training average against the total number of experts for a lower- and higher-complexity environments. The ratio of the position-segmentation factor to the beamforming-segmentation factor keeps 2:3 in training. Performance peaks at a moderate expert count for the complex setting with $N = 5$, $M = 3$, $K = 10$, indicating a trade-off where additional experts initially provide useful specialization but eventually incur higher routing complexity and reduced per-expert sample efficiency. For the simpler setting with $N = 3$, $M = 2$, $K = 6$, performance declines with more experts, suggesting overparameterization when task demands are modest.

Fig. 6 visualizes router gating activations produced by MoDE-based actor network under varying user positions and SNR conditions in environment $N = 3$, $M = 2$, $K = 6$ with base expert count of 2, position segmentation of 2, and beamforming segmentation of 3. Early in training, routing is concentrated on a few experts, e.g., 'pos3'≈ 0.287 and 'bf5'≈ 0.236, reflecting undeveloped specialization. Post-training, the router exhibits context-aware specialization: position experts provide broadly useful spatial encodings with more uniform activation, while beamforming experts develop distinct fingerprints correlated with specific SNR regimes or user layouts, e.g., user-position set 2 relies more on 'bf5' and 'bf6' after training, whereas set 4 shows near-uniform beamforming

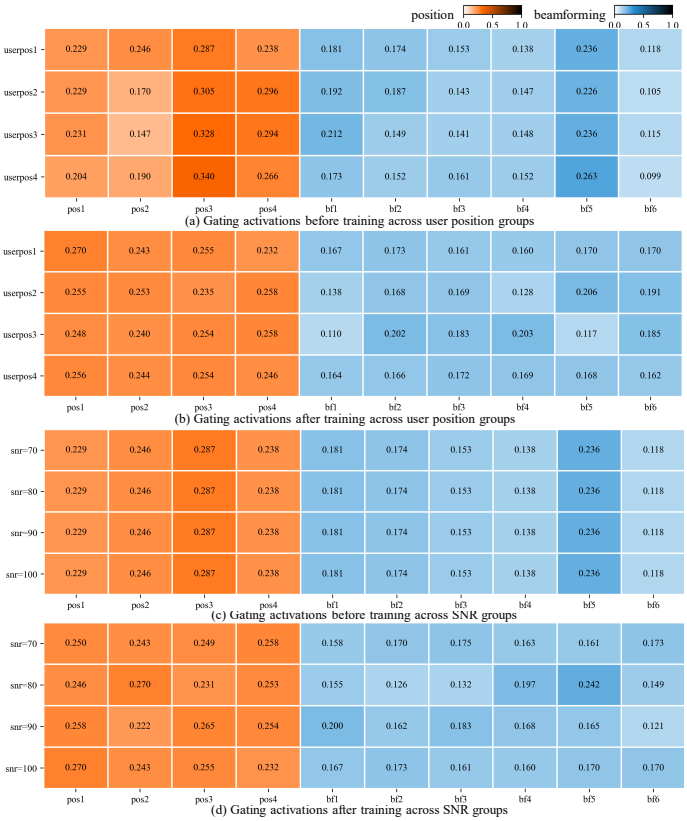


Fig. 6: Gating activations across environment settings of user-position and SNR variations.

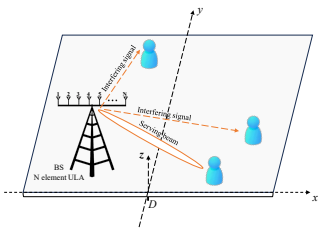


Fig. 7: MU-MIMO system model

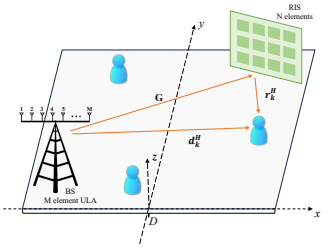


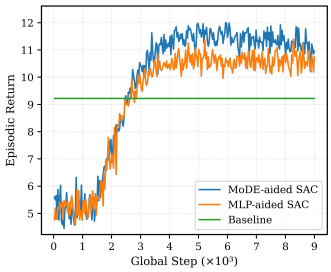
Fig. 8: RIS-aided MIMO system model

allocation. This demonstrates the adaptive, interpretable task decomposition and functional specialization of MoDE.

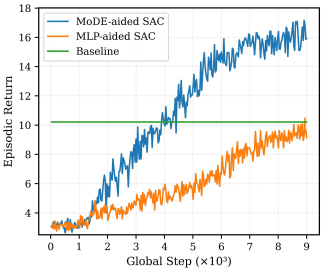
B. Joint Power Allocation and Beamforming in MU-MIMO

1) MoDE-Aided SAC In MU-MIMO System Model

We consider a downlink MU-MIMO setup depicted in Fig. 7, where a BS with N antennas serves K users distributed in a $D \times D\mathrm{m}^2$ area. Operating at $f_c = 28$ GHz with a path-loss exponent of 2.5, the aggregate channel matrix follows a deterministic LoS model comprising large-scale pathloss, distance-dependent phase delay, and array steering response. Beamforming follows the same practical design used for PASS with the direct physical channel as the equivalent channel. The optimization objective is to maximize SE as defined in Sec. III-A1. The MoDE architecture employs a layered expert structure to handle coupled decisions. The first layer outputs a K-dimensional power allocation factor from the state s containing channel state information and user positions; the second layer takes the enriched state $[s; a_{\text{power}}]$ to produce beamforming latent z_w. The continuous action a encodes both power allocation and beamforming parameters. The reward r is $r_t = 10\Big(\mathrm{SE}_t + 0.1\big(\mathrm{SE}_t - \mathrm{SE}_{t-1}\big)\Big)$ each step.

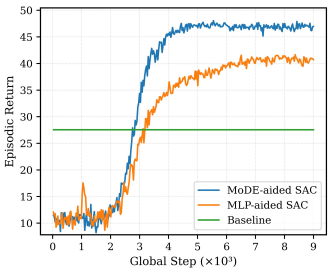


(a) N = 11, K = 18, D = 40

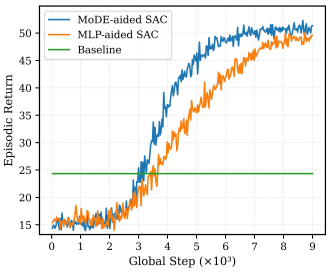


(b) N = 7, K = 13, D = 50

Fig. 9: Comparison of average return versus training steps in MU-MIMO.



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



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

Fig. 10: Comparison of average return versus training steps in RIS-aided MIMO.

2) Experimental Results

Fig. 9 compares the average return during training for MoDE-aided SAC, MLP-aided SAC, and a heuristic baseline using Minimum Mean-Square Error (MMSE) precoding with water-filling power allocation. Results show when $N = 7$, $K = 13$, $D = 50$ and $N = 11$, $K = 18$, $D = 40$, MoDE-aided SAC achieves faster convergence and a higher final performance compared to MLP-aided SAC, with both significantly surpassing the baseline after sufficient training.

C. Joint RIS Phase and Beamforming in RIS-Aided MIMO

1) MoDE-Aided SAC In RIS-Aided MIMO System Model

We examine a RIS-aided MIMO downlink as illustrated in Fig. 8, where a BS equipped with M transmit antennas serves K users distributed in a $30\times30\,\mathrm{m}^2$ region, through both direct links and reflections from an N-elements RIS. 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 RIS-reflected paths. The reference path loss is set to $P_{lo} = -20$ dB at $d_o = 1$ m. The composite channel for user k is $\mathbf{h}_k^H = \mathbf{r}_k^H \mathbf{\Theta} \mathbf{G} + \mathbf{d}_k^H$, where $\mathbf{G}$ is the BS-to-RIS link, $\mathbf{r}_k$ the RIS-to-user k link, $\mathbf{d}_k$ the direct BS-to-user k link, and $\Theta = \mathrm{diag}(e^{j\theta_1}, \ldots, e^{j\theta_N})$ is the RIS phase-shift matrix. We use the same beamforming design in MU-MIMO using effective channel constructed from $\mathbf{h}_k^H(\Theta)$. our objective of maximizing the total SE is defined the same as Sec. III-A1. In MoDE-based actor network, the first layer output both sine and cosine components of phase shifts from state s including previous RIS phases, effective channel, and user locations; the second layer uses the enriched state $[s; a_{\text{phase}}]$ to produce beamforming latent z_w for precoder optimization. The continuous action a encodes both RIS phase parameters and beamforming latent variables. The reward r is $r_t = 10\Big(\mathrm{SE}_t + 0.1\big(\mathrm{SE}_t - \mathrm{SE}_{t-1}\big)\Big)$ each step.

2) Experimental Results

Fig. 10 presents the average return versus training steps under $N = 24$, $M = 2$, $K = 3$, and $N = 36$, $M = 4$, and $K = 6$. The baseline operates on local copies of environment channels and alternately updates The results show that MoDE-aided SAC demonstrates faster convergence and higher final performance compared to MLP-aided SAC. Both learning-based methods substantially outperform a conventional alternating optimization baseline using alternate MMSE precoder and closed-form phase updates after sufficient training.

IV. Conclusion

We proposed MoDE-aided DRL framework for tightly coupled, high-dimensional optimization in wireless networks to enhance system performance. It enables effectively decoupling, specializing, and coordinated adaptation of both system configuration and signal parameters using real-time, sample-efficient feedback. This capability is validated across three representative scenarios, demonstrating effective joint optimization in each case. Furthermore, the proposed architecture naturally extends to other coupled wireless problems. Overall, our methodology highlights the potential of combining MoDE architectures with DRL for scalable, real-time resource optimization in 6G wireless systems.

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] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, "Adaptive mixtures of local experts," *Neural computation*, vol. 3, no. 1, pp. 79–87, 1991.

[8] M. I. Jordan and R. A. Jacobs, "Hierarchical mixtures of experts and the em algorithm," *Neural computation*, vol. 6, no. 2, pp. 181–214, 1994.

[9] 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.

[10] 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.

[11] Y. Zhou, T. Lei, H. Liu, N. Du, Y. Huang, V. Zhao, A. M. Dai, Q. V. Le, J. Laudon *et al.*, "Mixture-of-experts with expert choice routing," *Advances in Neural Information Processing Systems*, vol. 35, pp. 7103–7114, 2022.

[12] 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.

[13] 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.