Response to the Editors and Reviewers

Paper ID: MM-025164

Paper Title: Plasticity-Aware Mixture of Experts for Learning Under QoE Shifts in Adaptive Video Streaming

Dear Editors and Reviewers:

We sincerely appreciate the opportunity to respond to your and the reviewers' comments and improve the quality of our manuscript. We are submitting the revised version of our manuscript and want to express our gratitude for the constructive comments provided by you and the reviewers.

In response to your and the reviewers' feedback, we have carefully revised the manuscript, incorporating all necessary amendments, which are indicated in blue throughout the document. We have also provided detailed explanations for each comment in the following part of this reply letter. We hope that these revisions and our responses can address your comments and those of the reviewers.

Thank you very much once again for your invaluable comments to improve our research.

Yours sincerely,

Zhiqiang He and Zhi Liu.

1. Response to the comments from Associate Editor

• ***Comment 1: The reviewers have a number of significant concerns about the paper. Some are in terms of presentation, i.e., that the framework and algorithm should be presented more formally to the experiments, where the reviewers find that comparisons should be made to other, learning-based approaches.***

Response: Thank you very much for your valuable feedback. We sincerely appreciate your constructive suggestions regarding both the presentation and the experimental comparisons.

1. Regarding the presentation of the framework and algorithm

1) We have moved the experimental setup from the appendix to the main text.

2) We now provide a formal description of the algorithm, including detailed pseudocode.

3) We have added clear explanations of the algorithm's inputs and outputs.

The narrative structure and experimental logic of our paper are as follows:

Framework for Algorithms: We address the QoE Shift problem in adaptive bitrate streaming (ABR). Unlike prior methods that rely on handcrafted priors, we analyze a model's internal states to study network plasticity, thereby removing the need for prior assumptions. We first empirically verify that MLP-based controllers in ABR exhibit plasticity loss (Fig. 1: Action Response to System Change in the revised manuscript). Building on this diagnosis, we propose PA-MoE—a Plasticity-Aware Mixture-of-Experts—which intuitively leverages multiple specialized experts to handle different regimes. To our knowledge, this is the first approach in ABR that tackles QoE Shift explicitly from the perspective of plasticity. Accordingly, we devote more space to demonstrating and quantifying plasticity loss, and we intentionally adopt a simple mitigation to underscore the substantial headroom available when optimizing ABR through this lens.

Experiments: In our experiments, we explain why PA-MoE alleviates plasticity loss, detail how experts are selected internally, characterize the ABR system's response dynamics, and decompose overall QoE gains by component. Analyses at both the algorithmic and system levels highlight the large, untapped opportunity for plasticity-oriented ABR design. To assess the algorithm's robustness and efficacy, we also conduct sensitivity analyses with respect to the learning rate and noise intensity. We also present comparisons with both learning-based and non-learning-based methods.

Modified:

1) In **Section V-A Experiment Setting**, we add the content:

The experiments were performed on a system equipped with an Intel(R) Core(TM) i5-10400 CPU @ 2.90GHz, without GPU acceleration. The set of available bitrates is defined as $\mathcal{A} = \{300, 750, 1200, 1850, 2850, 4300\}$ kbps. Each video segment has a duration of 4 seconds. The playback buffer can hold up to 60 seconds of content, and the video consists of 49 segments in total. PPO hyperparameters are detailed in Table 2. For each training run, the agent is trained for approximately two hours, with a total of 2 million timesteps and 1000 iterations. All hyperparameters, including random seeds, are kept consistent across different algorithms. All QoE shift patterns in this paper follow the format shown in Figure 1.

Table 1: Potential Hyperparameter Configurations For PPO.

Hyperparameter	**Value**
Learning Rate	1e-4
Batch size	2000
Minibatch Size	62
Number of Iteration	1000
Rollout steps per iteration	2000
Total Timesteps	2e6
Update Epochs	5
GAE-γ	0.99
GAE-λ	0.95
Clip ϵ	0.2
Entropy Coefficient	0
Value Function Coefficient	5
Activation	ReLu
Environment	{D, L, N}
# Experts	3
Expert Hidden Size	18
MoE	{MoE, SMoE, PA-MoE}
Router	{Top-K-Router, Softmax-Router}
Number of Selected Experts	1
Actor MoE	True
Critic MoE	True

Network Trace Datasets: In our experiments, we draw on three distinct sources of throughput traces: (i) recordings from HSDPA-based 3G networks [1], collected while smartphones streamed video during travel on subways, trams, trains, buses, and ferries; (ii) the FCC corpus [2], created by stitching together randomly sampled logs from the "Web browsing" class in the August 2016 public release; and (iii) the Puffer open dataset [3], which comprises on-demand video sessions observed over heterogeneous access technologies, including wired links, Wi-Fi, and cellular networks (3G/4G/5G). The training set contains 127 traces and the test set contains 142; there is no overlap between them. These datasets will be released together with the source code upon publication.

2) In **Section Appendix F: Algorithm Pseudocode**, we add the content:

To facilitate implementation, Algorithm 1 provides complete pseudocode with explicit definitions of routing and plasticity-aware noise update rules.

Algorithm 1: PPO with Plasticity Aware MoE

Input: Env env; policy params π_ω (router + actor experts); value params V_ω (router + critic experts); rollout length T; update epochs K.

Output: Optimized policy $\boldsymbol{\omega}$

for *iteration* $= 1, 2, \ldots$ **do**

  Reset trajectory buffer; $\mathcal{B} \leftarrow \varnothing$

  for $t = 1, \ldots, T$ **do**

    Observe state s_t;

    Compute router features $h_t^{\pi_\omega} \leftarrow f_{\text{router}}^{\pi_\omega}(s_t)$;

    Compute clean logits $u^{\pi_\omega} \leftarrow W_{\text{topk}}^{\pi_\omega} h_t^{\pi_\omega}$;

    Add router noise scale z^{π_ω};

    $n^\star \leftarrow \arg\max_j \left(u_j^{\pi_\omega} + z_j^{\pi_\omega}\right)$; `# Top-1 expert index`

    $e_t^{\pi_\omega} \leftarrow \text{Expert}_{n^\star}^{\pi_\omega}(s_t)$; `# selected actor expert`

    $\ell_t \leftarrow W_{\text{policy}} e_t^{\pi_\omega}$; `# action logits`

    Sample $a_t \sim \text{Categorical}(\text{logits} = \ell_t)$ and save $\log \pi^\omega(a_t|s_t)$;

    Compute value $h_t^{V_\omega}$, u^{V_ω}, z^{V_ω} analogously;

    $m^\star \leftarrow \arg\max_j \left(u_j^{V_\omega} + z_j^{V_\omega}\right)$;

    $e_t^{V_\omega} \leftarrow \text{Expert}_{m^\star}^{V_\omega}(s_t)$;

    $v_t \leftarrow W_{\text{value}} e_t^{V_\omega}$;

    Step env with a_t to get r_t, s_{t+1}, d_{t+1};

    Push $(s_t, a_t, r_t, d_{t+1}, \log \pi_\theta(a_t|s_t), V_t)$ into $\mathcal{B}$;

    if d_{t+1} **then**

      reset env and continue

  Compute advantages A_t and returns R_t from $\mathcal{B}$;

  for *epoch* $= 1, \ldots, K$ **do**

    for *each minibatch* $B \subset \mathcal{B}$ **do**

      Update $\boldsymbol{w}$ using noise injection by Eq. (11) in the revised manuscript;

  (optional) Save checkpoints periodically;

3) In **Section II-A Problem Formulation**, we add the content:

The state and action spaces are aligned with mainstream approaches such as Pensieve [4]. The environment follows a standard reinforcement learning formulation: the agent interacts with the streaming system by observing network and playback states and selecting bitrate actions accordingly. The input is the state $s_t \in \mathbb{R}^{6\times 8}$, comprising six categories of information, each tracked over the most recent eight time steps: (i) normalized last selected bitrate; (ii) normalized buffer occupancy; (iii) measured throughput (downloaded size per unit time); (iv) normalized delay; (v) sizes of the next video chunk for all bitrate levels; and (vi) remaining number of chunks until the end of the video. PA-MoE outputs a discrete bitrate action $a_t \in \{0, 1, 2, 3, 4, 5\}$, where each value corresponds to one of the six available bitrate levels in the streaming system.

2. Concerning the comparative experiments

We have expanded the evaluation to include comparisons with both non-learning-based approaches (Buffer-Based [5], Rate-Based [6], and RobustMPC [7]) and learning-based approaches (Pensieve [4] and Merina [2], a meta-learning method). The results are as follows:

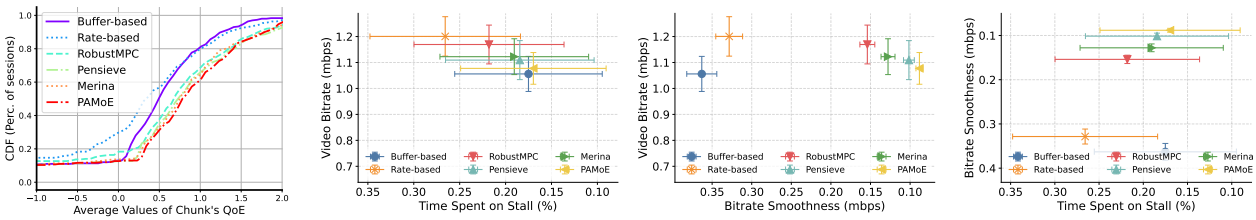

Figure 1: Comparing PA-MOE with recent ABR algorithms over the Train set.

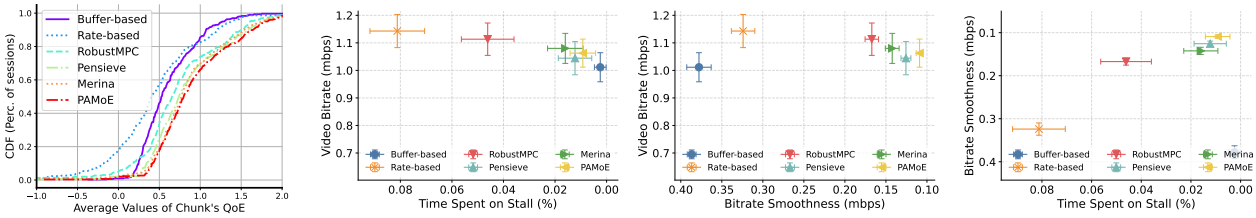

Figure 2: Comparing PA-MOE with recent ABR algorithms over the Test set.

The results—illustrated by the cumulative distribution function (CDF) plot, where curves farther toward the lower right indicate better performance—show that PA-MoE consistently outperforms the baselines, demonstrating robustness and effectiveness even against approaches that leverage prior knowledge (e.g., the meta-RL method Merina). Pairwise comparisons of QoE components—video bitrate, stall time, and smoothness—indicate that the gains primarily stem from improved smoothness and reduced stall time.

2. Response to Reviewer 1

• ***Comment 1: I noticed that several key studies have not been included, although they are directly relevant to the discussion on improving Quality of Experience (QoE) in adaptive video streaming. For example, the work on "Smart algorithm in wireless networks for video streaming based on adaptive quantization" (Concurrency and Computation: Practice and Experience, 2023) presents an intelligent approach that reduces resource consumption while maintaining high streaming quality. Similarly, "An automated model for the assessment of QoE of adaptive video streaming over wireless networks" (Multimedia Tools and Applications, 2021) provides a systematic framework for evaluating QoE metrics, offering valuable insights for measuring performance in real-world conditions. Furthermore, the study "A QoE adaptive management system for high definition video streaming over wireless networks" (Telecommunication Systems, 2021) introduces a management system that dynamically balances network resources to enhance user-perceived quality. Incorporating these contributions would strengthen the manuscript by providing a more comprehensive comparison with prior research, while also showing how other authors have addressed the dual challenge of minimizing resource usage and improving application-level QoE.***

Response:

We sincerely thank the reviewer for this valuable suggestion. We have cited the three papers mentioned in the **Introduction** of the manuscript; they correspond to references [5]–[7]. We first briefly summarize these works and then analyze how they differ from the present study. Finally, we include the corresponding revisions made to the manuscript.

The three Papers you mentioned:

1. "Smart algorithm in wireless networks for video streaming based on adaptive quantization" (Concurrency and Computation: Practice and Experience, 2023), which proposes an adaptive quantization strategy to dynamically adjust QP parameters according to network conditions, thus maintaining perceptual quality while reducing bandwidth usage. [Cited as [5] in the paper]
2. "An automated model for the assessment of QoE of adaptive video streaming over wireless networks" (Multimedia Tools and Applications, 2021), which introduces an automated machine-learning-based framework for real-time QoE evaluation using QoS and content features under wireless network fluctuations. [Cited as [6] in the paper]
3. "A QoE adaptive management system for high definition video streaming over wireless networks" (Telecommunication Systems, 2021), which presents a QoE-driven management mechanism that adapts encoding and transmission parameters to balance network utilization and perceived video quality. [Cited as [7] in the paper]

How our work differs:

This addition acknowledges recent progress in system- and encoding-level QoE optimization while clarifying how our work differs from these studies. Specifically, we focus on a learning-level challenge: when QoE objectives shift dynamically across heterogeneous video types and user contexts, conventional models suffer from plasticity loss and

fail to adapt. The proposed Plasticity-Aware Mixture of Experts (PA-MoE) addresses this limitation by maintaining adaptability under continuously evolving QoE optimization goals. We hope these revisions improve the manuscript's clarity and depth, and we sincerely appreciate your valuable feedback.

Revision:

We have revised the Introduction to include a concise discussion of these related works, as shown in the newly added text (highlighted in blue):

Recent research has sought to improve QoE from the *encoding*, *management*, and *assessment* perspectives, such as adaptive quantization for video encoding [8], QoE-driven management systems [9], and automated QoE evaluation models [10]. However, these approaches primarily target system-level adaptation and do not fully capture the evolving nature of QoE.

[8] M. Taha and A. Ali, "Smart algorithm in wireless networks for video streaming based on adaptive quantization," Concurrency and Computation: Practice and Experience, vol. 35, no. 9, p. e7633, 2023.

[9] M. Taha, A. Canovas, J. Lloret, and A. Ali, "A qoe adaptive management system for high definition video streaming over wireless networks," Telecommunication Systems, vol. 77, no. 1, pp. 63–81, 2021.

[10] M. Taha, A. Ali, J. Lloret, P. R. Gondim, and A. Canovas, "An automated model for the assessment of qoe of adaptive video streaming over wireless networks," Multimedia Tools and Applications, vol. 80, no. 17, pp. 26 833–26 854, 2021.

• ***Comment 2: The algorithm is described textually, without formal pseudo-code or stepwise presentation. This makes it difficult for readers to reproduce the exact update procedure, including noise injection and expert selection. Including explicit pseudo-code would improve clarity and reproducibility.***

Response:

Thank you for the valuable suggestion. We have added complete pseudocode for the proposed algorithm to improve clarity and reproducibility. Specifically, the noise-injection update is shown in Algorithm 2, line 23; expert selection is detailed in Algorithm 2, lines 9 and 14.

The Algorithm 2 provides a step-by-step description of PA-MoE, including (i) the routing mechanism that selects the top-1 expert based on noise-perturbed logits, (ii) the integration of plasticity-aware noise during policy and value updates, and (iii) the overall PPO training loop for optimization. This explicit formulation enables readers to trace how the MoE architecture interacts with PPO updates and how controlled noise injection maintains a balance between plasticity and stability.

The newly added section and algorithm are highlighted in blue (**Appendix F**).

To facilitate implementation, Algorithm 2 provides complete pseudocode with explicit definitions of routing and plasticity-aware noise update rules.

Algorithm 2: PPO with Plasticity Aware MoE

Input: Env env; policy params π_ω (router + actor experts); value params V_ω (router + critic experts); rollout length T; update epochs K.

Output: Optimized policy $\boldsymbol{\omega}$

for *iteration* $= 1, 2, \ldots$ **do**

  Reset trajectory buffer; $\mathcal{B} \leftarrow \varnothing$

  for $t = 1, \ldots, T$ **do**

    Observe state s_t;

    Compute router features $h_t^{\pi_\omega} \leftarrow f_{\text{router}}^{\pi_\omega}(s_t)$;

    Compute clean logits $u^{\pi_\omega} \leftarrow W_{\text{topk}}^{\pi_\omega} h_t^{\pi_\omega}$;

    Add router noise scale z^{π_ω};

    $n^\star \leftarrow \arg\max_j \left(u_j^{\pi_\omega} + z_j^{\pi_\omega}\right)$; `# Top-1 expert index`

    $e_t^{\pi_\omega} \leftarrow \text{Expert}_{n^\star}^{\pi_\omega}(s_t)$; `# selected actor expert`

    $\ell_t \leftarrow W_{\text{policy}} e_t^{\pi_\omega}$; `# action logits`

    Sample $a_t \sim \text{Categorical}(\text{logits} = \ell_t)$ and save $\log \pi^\omega(a_t|s_t)$;

    Compute value $h_t^{V_\omega}$, u^{V_ω}, z^{V_ω} analogously;

    $m^\star \leftarrow \arg\max_j \left(u_j^{V_\omega} + z_j^{V_\omega}\right)$;

    $e_t^{V_\omega} \leftarrow \text{Expert}_{m^\star}^{V_\omega}(s_t)$;

    $v_t \leftarrow W_{\text{value}} e_t^{V_\omega}$;

    Step env with a_t to get r_t, s_{t+1}, d_{t+1};

    Push $(s_t, a_t, r_t, d_{t+1}, \log \pi_\theta(a_t|s_t), V_t)$ into $\mathcal{B}$;

    if d_{t+1} **then**

      reset env and continue

  Compute advantages A_t and returns R_t from $\mathcal{B}$;

  for *epoch* $= 1, \ldots, K$ **do**

    for *each minibatch* $B \subset \mathcal{B}$ **do**

      Update $\boldsymbol{w}$ using noise injection by Eq. (11) in the revised manuscript;

  (optional) Save checkpoints periodically;

• ***Comment 3: The convergence analysis relies on L-smoothness and μ-strong convexity assumptions. Neural network losses under nonstationary QoE shifts are highly nonconvex, creating a potential gap between theory and real-world performance. The authors should discuss the limitations of these assumptions.***

Response: We thank the reviewer for this insightful and constructive comment.

1. **Nonconvex:** We fully agree that the assumptions of μ-strong convexity and L-smoothness are idealized and may not strictly hold for deep neural networks under nonstationary QoE objectives. However, in PPO-based methods, each policy update is constrained to a trust region, within which the loss can be well approximated by a local quadratic form. This local approximation motivates adopting assumptions such as L-smoothness and μ-strong to capture the objective's local behavior. Similar approximations have been employed in prior theoretical analyses, e.g., [11].

The intuition is that, although the overall optimization landscape is nonconvex, each update step is restricted to a trust region. Within this region, the objective behaves approximately convex, effectively reducing a nonconvex problem to a sequence of locally convex subproblems.

We have also explained this point in the paper (**Section IV-B**):

In PPO, the policy update is constrained within a trust region, which ensures that each update only makes small changes to the current parameters. Although the global objective may be highly nonconvex, within the trust region, the loss function can be well approximated locally by a quadratic function. This approximation allows us to make certain assumptions about the behavior of the loss function, such as being L-smooth and μ-strongly convex, which is also considered in [11]. These assumptions are useful for theoretical analysis and for understanding how the parameters adapt to a changing environment.

2. **Limitation of the Assumption:**

Our analysis assumes local L-smoothness and μ-strong convexity within each PPO update. This approximation may hold only in a small neighborhood; noisy gradients or large effective step sizes can move the iterate outside this region, invalidating the quadratic approximation.

In the revised manuscript, we have added a paragraph at the end of the **Section IV-B Training Experts with Plasticity Injection** subsection to explicitly discuss this limitation. The following clarification has been incorporated into the text:

While the convergence proof assumes L-smoothness and μ-strong convexity for analytical convenience, these are intended as local regularity conditions within each PPO step rather than claims of global convexity; when steps are large or gradients are indistinguishable from noise, the local quadratic approximation can break down. Nevertheless, PA-MoE exhibits stable convergence under nonconvex neural losses in practice, suggesting that its plasticity-aware updates remain effective beyond these idealized assumptions.

[11] A. Agarwal, S. M. Kakade, J. D. Lee, and G. Mahajan, "On the theory of policy gradient methods: Optimality, approximation, and distribution shift," Journal of Machine Learning Research, vol. 22, no. 98, pp. 1–76, 2021.

• ***Comment 4: The balance between active forgetting and memorization depends heavily on the learning rate η and noise strength γ. The manuscript lacks guidelines or sensitivity analysis for selecting these hyperparameters, which is critical for practical adoption.***

Response: We sincerely appreciate the reviewer's insightful comments. In the revised version, we have completed and strengthened the experiments for both parts.

1. **Sensitivity analysis for noise strength γ:**

We already conducted the sensitivity analysis of the noise strength γ in the subsection **Section V-G: Parameter Sensitivity Analysis in PA-MoE**.

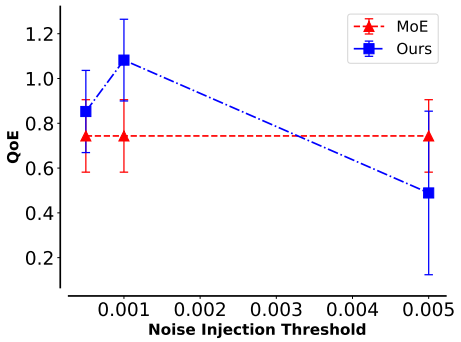

Figure 3: Comparison of QoE Methods using the IQM (Interquartile Mean) with a focus on the 25th-75th percentile returns. This approach aims to highlight the relative performance of different methods while minimizing the impact of outliers.

In PA-MoE, the magnitude of noise injection is a critical parameter. Our sensitivity analysis, presented in Fig. 3, reveals that both excessively low and high noise levels lead to performance degradation. This observation aligns with our theoretical analysis and validates Theorem 1. Furthermore, as shown in Appendix L, visualizations of the weight magnitudes indicate that the neural network adaptively adjusts its weights to retain memory and counteract noise-induced forgetting. These findings underscore the importance of carefully tuning the noise level to balance plasticity and stability in dynamic environments.

2. **Sensitivity analysis for learning rate η:**

We added a learning-rate sensitivity analysis to assess its effect on performance. The corresponding figures and explanations are included in the manuscript. Both excessively large and excessively small learning rates degrade performance, in agreement with our theoretical results. The added content is as follows:

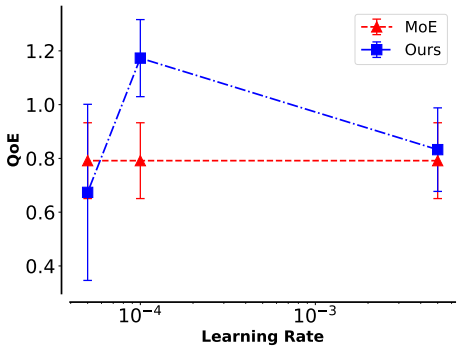

Figure 4: Comparison of QoE Methods using the IQM (Interquartile Mean) with a focus on the 25th-75th percentile returns.

Figure 4 shows QoE as a function of the learning rate. We find that both excessively large and excessively small

learning rates degrade performance, in agreement with our theoretical results. An inappropriate learning rate may even violate the trust-region assumption. Nevertheless, the experiments indicate that, even with a learning rate as large as 0.005—rare in practice—our method still achieves a modest improvement.

3. **Guidelines for practical**

In practical algorithm testing, we place greater emphasis on tuning the noise parameters. Our rationale is as follows:

In our theory, the update rule injects noise whose amplitude is scaled by the learning rate, i.e.,

$$w_{t+1} = w_t - \eta \nabla L_t(w_t) + \eta\gamma\epsilon_t.$$

So the effective plasticity control depends on the product $k = \eta\gamma$ (see Eq. (11) in the paper). Moreover, the tracking-error bound

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left\|e_t\right\|^2 \leq C\left(\eta\gamma^2 d + \frac{P_T^2}{T\eta}\right), \tag{1}$$

makes this coupling explicit: η multiplies the noise term, whereas $\frac{1}{\eta}$ governs the memorization term (Theorem 1).

Firstly, **Method-specific vs. optimizer-specific.** η is an optimizer-level hyperparameter shared by all methods (PPO with clipping/trust-region), rather than specific to PA-MoE. To ensure fair comparisons and to remain within the stability regime $\eta \leq \frac{1}{L}$, we use a fixed η across all models, following standard PPO practice. Sweeping η would simultaneously alter trust-region behavior and confound method comparisons.

Secondly, **Theory prescribes tuning $k = \eta\gamma$, practice varies γ with fixed η.** Because the effective injected noise is $\eta\gamma$, varying γ at fixed η directly explores the plasticity–stability trade-off predicted by the bound. Our noise-sensitivity analysis sweeps γ and shows the expected "too small/too large both hurt" behavior (Fig. 12 in the revised manuscript), identifying a robust band (e.g. $\gamma \in [0.0005, 0.005]$) where QoE is stable—precisely the regime in which k is well tuned.

Thirdly, **Actionable guideline.** In deployment, practitioners typically (i) set η based on optimizer stability and throughput constraints (as we do across all baselines), and then (ii) tune γ to target a good k. This procedure aligns with the theory (dependence on the product $\eta\gamma$) and with our ablations (the γ-sweep).

• ***Comment 5*: *The Top-1 expert selection may bias learning toward certain experts, and the paper does not analyze whether all experts are effectively utilized. Additionally, the computational cost of multiple experts with noise injection is not discussed, raising concerns about scalability.***

Response: We thank the reviewer for raising this important point regarding expert utilization and scalability. We address these points in turn.

1. Whether all experts are effectively utilized?

As shown in the subsection "**Section V-C: Are the selection probabilities for each expert balanced?**" (Fig. 5), all experts in PA-MoE are utilized uniformly, exhibiting a well-balanced routing distribution throughout training. In

this reply, we have included the result figure, as shown in Figure 5. The results in Figure 5 confirm that PA-MoE selects evenly across the three experts, achieving load balance.

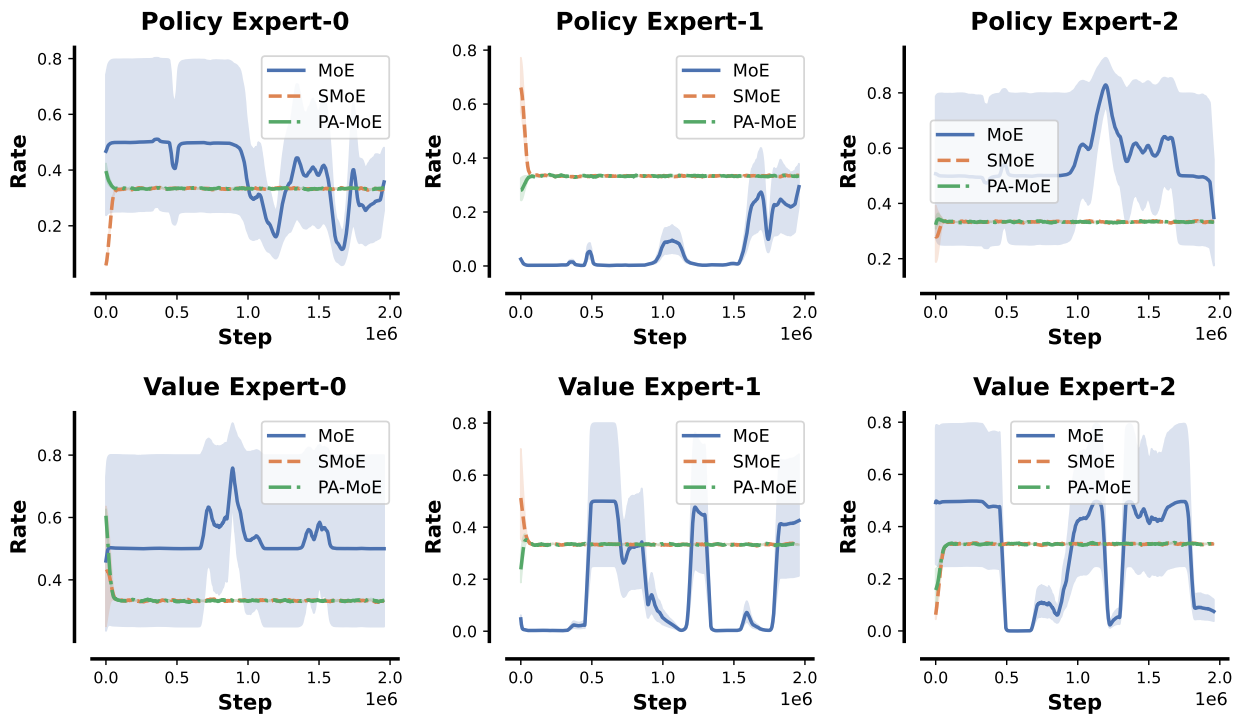

Figure 5: Probability distribution of each expert selected for Policy and Value.

2. Computational cost of multiple experts

Although PA-MoE contains multiple experts, only one expert (Top-1) is activated per input during both training and inference. Consequently, the per-sample computational cost scales with the number of activated experts (= 1) rather than the total number of experts. The additional overhead from noise injection is negligible, since it amounts to applying small Gaussian perturbations to expert parameters. Another source of computational overhead comes from the router, but it does not increase with the number of experts.

To make this explicit, we added the following clarification in the subsection **Section IV: Architectural Paradigm of the MoE Framework**:

Although the architecture includes N experts, only one expert is selected for each input, keeping the per-sample complexity at $O(1)$.

3. Response to Reviewer 2

3.1. Comments

This manuscript focuses on adaptive video streaming, with a specific emphasis on modeling and analyzing plasticity in multimedia delivery systems. The study directly addresses core multimedia topics such as content adaptation, video quality optimization, and real-time transmission under dynamic Quality of Experience (QoE).

Response: We sincerely thank the reviewer for the positive and thoughtful summary, as well as for recognizing the central focus of our work.

Indeed, this study explores adaptive video streaming from a learning-level perspective, with particular emphasis on modeling and maintaining plasticity in multimedia delivery systems under dynamically changing Quality of Experience (QoE) objectives.

We have addressed all the points you raised, including (i) the motivation for adopting MoE, (ii) the key design choices in PA-MoE, (iii) additional experiments that benchmark both learning-based and non-learning-based approaches, (iv) the proposed framework for describing the simulation setup, and (v) the connection between the theoretical analysis and the overall method.

• ***Comment 1: The manuscript aims to address the issue where differences in user profiles and video content lead to varying weights for QoE factors, resulting in user-specific QoE functions and thus different optimization objectives. While heterogeneous QoE has been explored in the Adaptive Bitrate (ABR) field—for instance, [R1] and [R2] utilize meta-learning and self-attention-based representation learning for personalized bitrate adaptation, and [R3] introduces spatial–temporal learning to enhance the generalization of Deep Reinforcement Learning (DRL)-based ABR—the application of Mixture of Experts (MoE) in adaptive video streaming remains relatively unexplored. Therefore, the motivation for employing MoE over existing techniques needs to be more clearly and compellingly articulated.***

Response: Thank you for the valuable comment. In response, we will address your points from two perspectives: (i) we will compare and clarify the differences between our work and the related studies you mentioned, and (ii) we will explain the motivation for adopting the Mixture-of-Experts (MoE) framework.

1. **The difference between our work and the related studies you mentioned**

We have cited the literature you mentioned and, in the revised paper, added the relevant comparisons and discussion. Our more detailed comparison and explanation are as follows:

R1: "Mansy: Generalizing neural adaptive immersive video streaming with ensemble and representation learning" trained a QoE identifier, meaning that it requires prior knowledge of how user QoE preferences change over time. In other words, the model introduces **explicit prior information** about preference dynamics to enhance generalization capability. [Cited as [40] in the paper]

R2: "A novel spatial-temporal learning method for enhancing generalization in adaptive video streaming" focuses on variations in network conditions. It trains a **network-bandwidth classifier** as prior knowledge to enhance the model's generalization capability. [Cited as [41] in the paper]

R3: "Merina+: Improving generalization for neural video adaptation via information theoretic meta-reinforcement learning," adopts a meta-reinforcement learning approach. **It compresses prior knowledge of network bandwidth through an autoencoder**, and then incorporates this latent representation as an input to the policy network to improve generalization. [Cited as [42] in the paper]

All three of the above studies follow a two-stage approach: in the first stage, prior knowledge is represented or encoded, and in the second stage, policy optimization is performed based on the learned representation. For this class of methods, we have provided a detailed comparison and discussion in the **Section I: Introduction**, as shown below.

However, two-stage optimization methods encounter several challenges that hinder effective QoE optimization. Inaccuracies in user-level predictions can cascade into subsequent stages, resulting in suboptimal outcomes. Furthermore, their sequential nature, dependence on pre-trained models, and reliance on historical data delay real-time adjustments and updates, while also limiting the ability to generalize when faced with QoE variations beyond the training distribution. Existing approaches for handling dynamic QoE weights, such as multi-objective Q-networks [12] and meta-reinforcement learning [2], typically rely on a key assumption: the dynamics of the weight changes are known in advance and can be used either as input to the model [12] or for pretraining [2]. In contrast, our method removes this critical assumption by directly addressing the problem from the perspective of plasticity, making our approach more aligned with realistic industrial scenarios.

Based on the references you mentioned, we have added corresponding comparisons and discussions to the **Section III: Related Work**, as shown below.

For example, some methods first employ a monitor to detect system changes before applying adjustments such as retraining [13]; others train a QoE identifier [14] and a network-bandwidth classifier [15]; and in meta-reinforcement learning, an autoencoder is used to compress prior knowledge about network bandwidth [16]. Meta reinforcement learning enables systems to "learn how to learn" across multiple tasks [17, 18], facilitating rapid adaptation. However, this approach often assumes that tasks share certain stationary properties in their distributions, which is often unrealistic in real-world adaptive video streaming applications.

2. **The motivation behind our adoption of the Mixture-of-Experts (MoE) framework.**

Reason for adopting the MoE framework: Most existing methods for handling dynamic objectives adopt a two-stage optimization scheme: in the first stage, prior knowledge about QoE dynamics is encoded and supplied to the policy network; in the second stage, the policy is optimized. However, in industrial deployments, acquiring and validating such priors is highly resource-intensive, often requiring substantial human effort and data collection. Once prior knowledge is introduced, it implicitly assumes a particular distribution. When the underlying dynamics shift beyond this distribution (i.e., out of distribution), the model fails to generalize, rendering the system difficult to deploy. We therefore propose a single-stage optimization method that introduces no prior knowledge, relying instead on the inherent properties of neural networks. Given that Mixture-of-Experts (MoE) leverages multiple experts to address different subproblems, it is a natural choice for modeling dynamic objectives.

We added the following description to the Mixture of Experts section in the original manuscript (**Section III-C: Mixture of Experts**):

The MoE architectures utilizes a router mechanism to dynamically assign input states to specialized expert networks for decision-making [19]. This approach has been widely adopted in large-scale models [20, 21, 22] and has shown exceptional performance in multi-task learning and Continual Learning (CL) scenarios [23]. Therefore, it is natural and intuitive to consider applying it to handle the objective shifts in AVS.

During deployment, we observed that applying Mixture-of-Experts (MoE) to adaptive video streaming can suffer from a loss of plasticity. We therefore conducted an in-depth analysis of this phenomenon, offering a new perspective on optimizing adaptive video streaming.

References for this Comment

[2] N. Kan, Y. Jiang, C. Li, W. Dai, J. Zou, and H. Xiong, "Improving generalization for neural adaptive video streaming via meta reinforcement learning," in Proceedings of the 30th ACM International Conference on Multimedia, 2022, pp. 3006–3016.

[12] A. Abels, D. Roijers, T. Lenaerts, A. Nowé, and D. Steckelmacher, "Dynamic weights in multi-objective deep reinforcement learning," in International conference on machine learning. PMLR, 2019, pp. 11–20.

[13] L. Zhang, G. Gao, and H. Zhang, "Towards data-efficient continuous learning for edge video analytics via smart caching," in Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems, 2022, pp. 1136–1140.

[14] D. Wu, P. Wu, M. Zhang, and F. Wang, "Mansy: Generalizing neural adaptive immersive video streaming with ensemble and representation learning," IEEE Transactions on Mobile Computing, 2024.

[15] G. Zhang, Z. Wang, H. Wei, M. Xiao, H. Yuan, D. Yu, and X. Cheng, "A novel spatial-temporal learning method for enhancing generalization in adaptive video streaming," IEEE Transactions on Mobile Computing, 2025.

[16] N. Kan, C. Li, Y. Jiang, W. Dai, J. Zou, H. Xiong, and L. Toni, "Merina+: Improving generalization for neural video adaptation via information-theoretic meta-reinforcement learning," IEEE Transactions on Circuits and Systems for Video Technology, 2025.

[17] S. Wang, J. Lin, and Y. Dai, "Mmvs: Enabling robust adaptive video streaming for wildly fluctuating and heterogeneous networks," IEEE Transactions on Multimedia, 2024.

[18] W. Li, X. Li, Y. Xu, Y. Yang, and S. Lu, "Metaabr: A meta-learning approach on adaptative bitrate selection for video streaming," IEEE Transactions on Mobile Computing, vol. 23, no. 3, pp. 2422–2437, 2023.

[19] W. Cai, J. Jiang, F. Wang, J. Tang, S. Kim, and J. Huang, "A survey on mixture of experts," arXiv preprint arXiv:2407.06204, 2024.

[20] J. Li, Z. Sun, X. He, L. Zeng, Y. Lin, E. Li, B. Zheng, R. Zhao, and X. Chen, "Locmoe: A low-overhead moe for large language model training." in IJCAI. ijcai.org, 2024, pp. 6377–6387.

[21] B. Lin, Z. Tang, Y. Ye, J. Cui, B. Zhu, P. Jin, J. Zhang, M. Ning, and L. Yuan, "Moe-llava: Mixture of experts for large vision-language models." CoRR, vol. abs/2401.15947, 2024.

[22] F. Xue, Z. Zheng, Y. Fu, J. Ni, Z. Zheng, W. Zhou, and Y. You, "Openmoe: an early effort on open mixture-of-experts language models," in Proceedings of the 41st International Conference on Machine Learning, ser. ICML'24. JMLR.org, 2024.

[23] H. Li, S. Lin, L. Duan, Y. Liang, and N. Shroff, "Theory on mixture-of-experts in continual learning," in The Thirteenth International Conference on Learning Representations, 2025. [Online]. Available: https://openreview.net/forum?id=7XgKAabsPp

• ***Comment 2: What is the key design in PA-MoE? What are the unique challenges of applying MoE to ABR? For example, the authors summarize in their second contribution that they incorporate controlled noise injection into experts to actively eliminate outdated knowledge, thereby maintaining the adaptive capability of the MoE framework. However, this design appears to be a general approach for MoE rather than one specifically tailored to ABR. Is this a novel contribution of this paper? How is this design connected to the specific problems in ABR?***

Response: We sincerely thank the reviewer for the insightful question regarding the key design of PA-MoE and the unique challenges of applying the Mixture-of-Experts (MoE) framework to adaptive bitrate (ABR) streaming. We address the questions point by point below.

1. **What is the key design in PA-MoE?**

The core design of PA-MoE is a *plasticity-preserving MoE* tailored for ABR's nonstationary objectives. Concretely, we introduce *controlled stochastic perturbations of expert weights* (rather than gating noise) to periodically refresh stale parameters and sustain neuron/expert utilization under distribution shift. This mechanism is intentionally simple and lightweight, but is guided by our analysis of plasticity loss in MoE: it maintains gradient diversity, prevents premature parameter dormancy, and preserves the ability to adapt as network and user conditions evolve. In addition, we reformulate a previously two-stage pipeline [14, 15, 16] into a *single-stage* procedure that removes the need for priors, improving deployability.

2. **What are the unique challenges of applying MoE to ABR? Is this a novel contribution of this paper?**

ABR-specific challenge. Unlike stationary tasks, ABR faces continual, goal-driven nonstationarity: the QoE objective shifts with throughput, latency, buffer dynamics, and user behavior. In this setting, standard MoE variants tend to suffer *loss of plasticity*: experts become under-utilized or "dormant," routing adapts more slowly, and the model overfits to recent regimes—limiting real-world QoE.

Our contributions. (i) We identify and analyze loss of plasticity in MoE under ABR dynamics and show its impact on QoE variability. (ii) We propose a weight-space noise mechanism that targets the plasticity failure mode at the neuron/expert level. This differs from common MoE variants that inject noise in the gating network (which mainly perturbs routing rather than rejuvenating expert parameters). (iii) We provide a *single-stage, no-prior* training formulation that simplifies deployment compared with two-stage designs.

While noise as a general idea is known, our novelty lies in (a) diagnosing the ABR-specific plasticity problem in MoE, (b) targeted application at expert weights to actively forget outdated knowledge, and (c) demonstrating that this plasticity lens enables a simpler, prior-free, single-stage solution that improves ABR QoE. We believe these insights are relevant beyond ABR, but our evidence and design choices are grounded in ABR.

3. **How is the design connected to the specific problems in ABR?**

All of our experiments, analyses, and conclusions are grounded in ABR; this, in our view, is the strongest connection. In ABR, QoE objectives are nonstationary, varying with both network conditions and user behavior. Existing

methods typically inject prior knowledge (e.g., learned QoE models or bandwidth predictors) to guide adaptation, which can lead to overfitting and limited generalization in real-world deployments. PA-MoE addresses this issue from a learning perspective, enabling the system to retain adaptability through intrinsic plasticity rather than external priors. Thus, although noise injection is conceptually simple, its role here is novel: it directly **tackles the plasticity degradation problem** specific to ABR's dynamically evolving objectives.

• ***Comment 3: In the simulations, only MoE and SMoE are compared. While this demonstrates that the proposed MoE designs improve performance relative to other MoE-based methods in ABR, it does not sufficiently validate or motivate the use of MoE in ABR solutions. Comparisons with other learning-based and non-learning-based benchmarks are necessary to highlight the superiority of MoE for ABR.***

Response: We sincerely thank the reviewer for this valuable and insightful comment. In the revised manuscript, we have expanded the experimental section to include comprehensive comparisons with both learning-based and non-learning-based benchmarks, thereby more rigorously validating the effectiveness of—and the motivation for—adopting the Mixture-of-Experts (MoE) framework in adaptive bitrate (ABR) streaming.

The newly added content is summarized as follows:

Section V-E: In this subsection, we compare our approach with methods that rely on prior knowledge, including learning-based approaches such as Pensieve [4] and the meta-learning method Merina [2], as well as non-learning-based approaches such as RobustMPC [7], RateBased [6], and BufferBased [5]. To ensure a fair and credible comparison, we use the same QoE-component coefficients and network architecture in Merina [16] for each of the experts in the MoE.

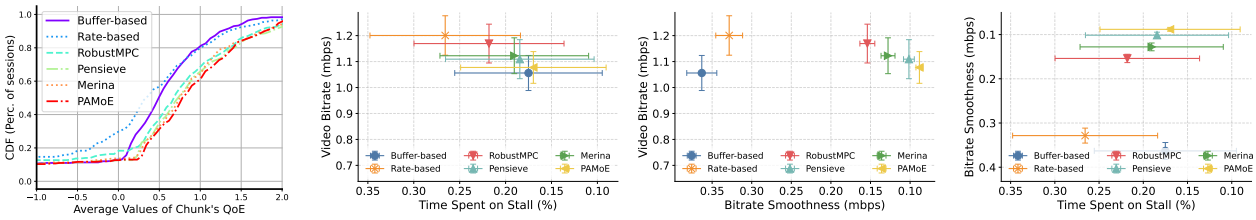

Figure 6: Comparing PA-MOE with recent ABR algorithms over the Train set.

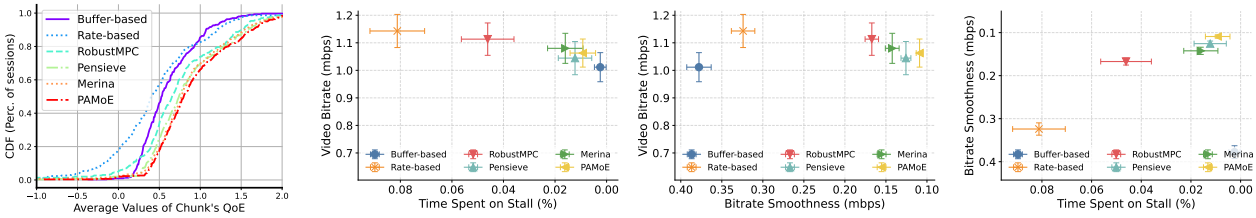

Figure 7: Comparing PA-MOE with recent ABR algorithms over the Test set.

Figure 6 presents the cumulative distribution functions (CDFs) of average QoE for all sessions and algorithms on the training set, along with pairwise comparisons of the QoE components—bitrate, smoothness, and stall time.

Figure 7 shows the corresponding results on the testing set. As shown, our proposed PA-MoE achieves state-of-the-art performance, even relative to the meta-learning method Merina, which leverages prior knowledge. These results underscore the substantial potential of optimizing adaptive bitrate (ABR) algorithms through the lens of plasticity.

• ***Comment 4: Furthermore, the simulation setup should be included in the main body of the paper rather than in the appendix to ensure reproducibility. It is also unclear which downlink network is considered in the simulations. Does the study employ Python simulations with real-world network traces (such as those used in Pensieve and Comyco), emulation, or a real-world testbed?***

Response: Thank you for your valuable comment. We address these points in turn.

1. The simulation setup should be included in the main body of the paper rather than in the appendix to ensure reproducibility.

Following your suggestion, we have moved the simulation setup to the "Experiment Setting" section in the main body of the paper.

Section V-A: Experiment Setting

The experiments were performed on a system equipped with an Intel(R) Core(TM) i5-10400 CPU @ 2.90GHz, without GPU acceleration. The set of available bitrates is defined as $\mathcal{A} = \{300, 750, 1200, 1850, 2850, 4300\}$ kbps. Each video segment has a duration of 4 seconds. The playback buffer can hold up to 60 seconds of content, and the video consists of 49 segments in total. PPO hyperparameters are detailed in Table 2. For each training run, the agent is trained for approximately two hours, with a total of 2 million timesteps and 1000 iterations. All hyperparameters, including random seeds, are kept consistent across different algorithms. All QoE shift patterns in this paper follow the format illustrated in Figure 1 in the revised manuscript.

Table 2: Potential Hyperparameters Configurations For PPO.

Hyperparameter	**Value**
Learning Rate	1e-4
Batch size	2000
Minibatch Size	62
Number of Iteration	1000
Steps	2000
Total Timesteps	2e6
Update Epochs	5
GAE-γ	0.99
GAE-λ	0.95
Clip ϵ	0.2
Entropy Coefficient	0
Value Function Coefficient	5
Activation	ReLU
Environment	{D, L, N}
# Experts	3
Expert Hidden Size	18
MoE	{MoE, SMoE, PA-MoE}
Router	{Top-K-Router, Softmax-Router}
Number of Selected Experts	1
Actor MoE	True
Critic MoE	True

2. It is also unclear which downlink network is considered in the simulations.

Following your suggestion, we have provided a more detailed explanation of the datasets we use in the "Experiment Setting" section; the added content is as follows:

Network Trace Datasets: In our experiments, we draw on three distinct sources of throughput traces: (i) recordings from HSDPA-based 3G networks [1], captured while smartphones streamed video during travel by subway, tram, train, bus, and ferry; (ii) the FCC corpus [2], created by stitching together randomly sampled logs from the "Web browsing" class in the August-2016 public release; and (iii) the Puffer open dataset [3], which comprises on-demand video sessions observed over heterogeneous access technologies, including wired links, Wi-Fi, and cellular systems (3G/4G/5G). A total of 127 trace files are included in the training dataset and 142 trace files in the testing dataset. There is no overlap between the training and testing datasets, and all datasets will be released together with the source code upon publication.

3. Does the study employ Python simulations with real-world network traces (such as those used in Pensieve and Comyco), emulation, or a real-world testbed?

Our experiments are conducted using Python-based simulations with real-world network traces, consistent with the setups employed in prior work such as Pensieve.

• ***Comment 5: How does the performance regret bound help the design of PA-MoE is not clear. In the current manuscript, the design part and the anlysis part looks de-coupled.***

Response: Thank you for this valuable and insightful comment. We address these points in turn.

1. How does the performance regret bound help the design of PA-MoE is not clear.

The performance regret bound is intended to provide theoretical justification for addressing QoE shift in PA-MoE by adding noise. Our analysis establishes the existence of a solution, and our experiments corroborate that improved solutions can indeed be found. How to further refine PA-MoE based on this theory is left for future work. For example, a promising direction is to inject noise selectively into dormant neurons rather than all neurons. Doing so would alter the update rule and, leveraging the current analysis, could yield a tighter bound—a step from theory to method. This would represent a progression from theory to method. That said, this paper focuses primarily on analysis: from a plasticity perspective, we show that there is substantial room to optimize for QoE shift, and we provide preliminary empirical and theoretical evidence for this claim.

Put differently, we present PA-MoE as a plasticity-oriented approach to addressing QoE shift, and we derive a performance (regret) bound that provides a formal guarantee. We did not adapt existing theories to guide the design of PA-MoE, nor did we aim to obtain a tighter bound.

2. In the current manuscript, the design part and the anlysis part looks de-coupled.

Our theoretical analysis is intended to show that the policy component we design admits a performance bound.

We provide an illustrative example to clarify the paper's aims and to explain how the theoretical analysis fits into the overall work.

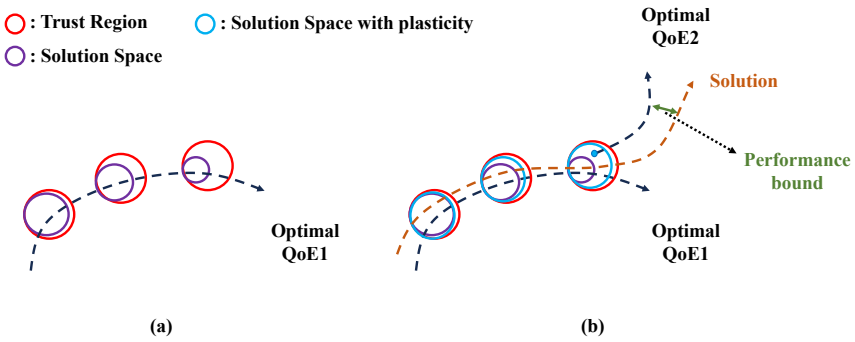

Figure 8: Schematic illustration of the theoretical analysis.

As illustrated in Figure 8 (a), a standard PPO update is performed within a trust region at each iteration. However, modern neural networks can suffer from loss of plasticity, whereby some neurons become inactive during training.

This lowers neuron utilization, effectively reducing the number of usable neurons and thereby shrinking the search space, as indicated by the purple circle in Figure 8 (a).

As shown in Figure 8(b), the blue circle denotes the solution space after noise injection. Because the perturbation mitigates loss of plasticity, the solution space expands. Consequently, when the QoE objective shifts, there is a larger region in which to search for parameter configurations capable of changing the current policy. In essence, PA-MoE addresses loss of plasticity in MoE rather than merely adding noise. Although many methods could alleviate this loss, we adopt the simplest possible approach to highlight the substantial headroom for optimization—under QoE shift in ABR—from the perspective of neural plasticity.

Our theoretical bound is derived under the assumption that each update is performed within a trust region. Because every iteration remains within this region [11], the bound we obtain for the PA-MoE solution also serves as a bound on the policy itself. The green "Performance bound" label in Figure 8(b) explicitly illustrates the relationship between the solution produced by our method and the optimal solution.

4. Response to Reviewer 3

4.1. Comments

Making bitrate decisions in adaptive bitrate streaming is critical for optimizing user QoE. This manuscript adopts a Mixture of Experts (MoE) framework to address the plasticity loss problem in current learning methods. The proposed PA-MoE balances memory retention with selective forgetting by leveraging the MoE architecture and noise injection. The theoretical analyses are solid and comprehensive, and the experiments demonstrate the effectiveness of the designed scheme.

Response: We sincerely thank the reviewer for the positive and encouraging evaluation of our work. We are pleased that the reviewer recognizes the significance of addressing plasticity loss in learning-based adaptive bitrate streaming and acknowledges both the soundness of our theoretical analysis and the effectiveness of our experimental results. We greatly appreciate this affirmation, which encourages us to further refine and extend the proposed PA-MoE framework in future research.

• ***Comment 1: Figure 1 aims to validate the effectiveness of MoE in adapting to QoE shifts. However, using bitrate as a metric is confusing. What is the rationale for selecting bitrate over QoE for plasticity evaluation? Additionally, please provide a detailed illustration of the QoE shift phenomenon depicted in the figure.***

Response: We sincerely thank the reviewer for this insightful and thought-provoking comment. The reviewer's observation regarding the choice of bitrate as the evaluation metric versus QoE directly points to the conceptual essence of our framework, and we truly appreciate this deep reflection.

1. Rationale for Selecting Bitrate over QoE for Plasticity Evaluation.

Rationale for Not Using QoE as the Evaluation Metric: The changes we introduce lie in the objective function itself, namely, QoE. If QoE were used directly as the plasticity metric, it would be difficult to determine whether variations in QoE arise from changes in the objective's parameters or from the adaptation capability of the learning-based method.

Rationale for Using Bitrate as the Evaluation Metric: QoE is not suitable for measuring plasticity. We therefore identify an internal system metric to assess the neural network's learning ability. The policy's response—i.e., its actions—provides an intuitive proxy for how the policy network evolves during the QoE shift. Accordingly, we use bitrate to quantify loss of plasticity.

2. Please provide a detailed illustration of the QoE shift phenomenon depicted in the figure.

In the revised manuscript, we have added illustrations to describe the QoE-shift phenomenon and to explain why we do not use QoE as a metric for plasticity (**Appendix D**):

In this subsection, we explain why we use the mean of the action outputs as the metric for evaluating the MLP's inability to handle QoE shifts. When the objective coefficients change abruptly, overall QoE may fluctuate even if the MLP's output actions remain unchanged, as shown in Fig. 9. Such QoE fluctuations are uninformative. For example, if the coefficient of the bitrate term in the QoE function changes from 1 to 6, the QoE contribution of bitrate rises sixfold, even though the policy is effectively unresponsive. Therefore, during objective-function shifts, performance

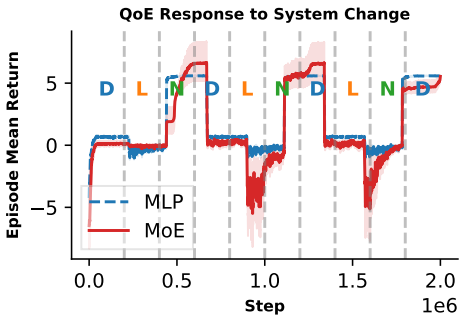

Figure 9: QoE output variation across different network architectures under shifted QoE reward conditions. D, L, and N represent distinct QoE metrics.

cannot be judged solely by QoE values, as they may convey a false sense of progress. This phenomenon may partly explain why many learning-based methods struggle to generalize under dynamically changing QoE objectives.

Therefore, when the objective function changes, comparing the QoE values for an MLP with unchanged outputs is uninformative, as the policy is no longer genuinely responsive.

We would like to once again thank the reviewer for raising this profound and insightful question.

• ***Comment 2: Please provide a clear explanation of the actual inputs and outputs of PA-MoE.***

Response: We sincerely thank the reviewer for this helpful question. We agree that clarifying the actual inputs and outputs of PA-MoE is essential for better understanding the model design. We added the following clarification at the end of the Problem Formulation subsection (**Section II-A Problem Formulation**):

The state and action spaces are aligned with mainstream approaches such as Pensieve [4]. The environment follows a standard reinforcement learning formulation: the agent interacts with the streaming system by observing network and playback states and selecting bitrate actions accordingly. The input is the state $s_t \in \mathbb{R}^{6\times 8}$, comprising six categories of information, each tracked over the most recent eight time steps: (i) normalized last selected bitrate; (ii) normalized buffer occupancy; (iii) measured throughput (downloaded size per unit time); (iv) normalized delay; (v) sizes of the next video chunk for all bitrate levels; and (vi) remaining number of chunks until the end of the video. PA-MoE outputs a discrete bitrate action $a_t \in \{0, 1, 2, 3, 4, 5\}$, where each value corresponds to one of the six available bitrate levels in the streaming system.

The input features and the output action space are kept consistent with those of Pensieve [4] and the Merina method [2].

• ***Comment 3: As PA-MoE is a learning-based ABR scheme, it should be compared against state-of-the-art reinforcement learning-based ABR methods. Have you considered comparing with representative baselines such as MPC, Pensieve, BOLA, Oboe, Pytheas, or Soda?***

Response: Thank you for your valuable comment. In the revised version of our manuscript, we have added comparisons with existing learning-based and non-learning-based methods. Our method outperforms the non-learning-based

method Robust MPC [7], the classic learning-based Pensieve [4], and the meta-reinforcement learning method Merina [2]. The added content is as follows (**Section V-E**):

In this subsection, we compare our approach with methods that rely on prior knowledge, including learning-based approaches such as Pensieve [4] and the meta-learning method Merina [2], as well as non-learning-based approaches such as RobustMPC, RateBased, and BufferBased [7]. To ensure a fair and credible comparison, we use the same QoE-component coefficients and network architecture in Merina [16] for each of the experts in the MoE.

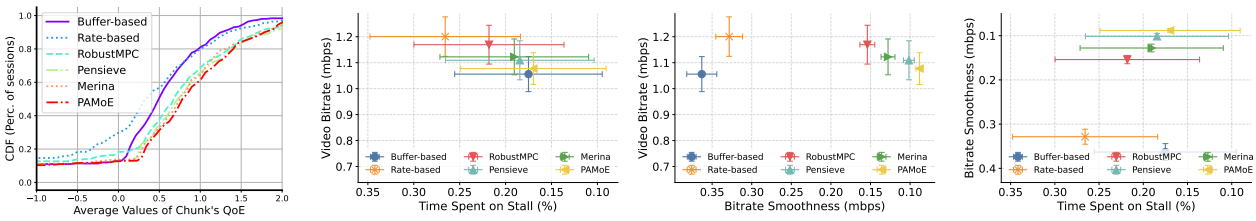

Figure 10: Comparing PA-MOE with recent ABR algorithms over the Train set.

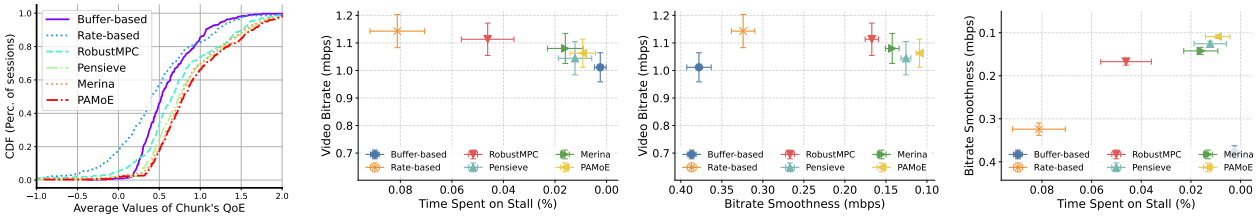

Figure 11: Comparing PA-MOE with recent ABR algorithms over the Test set.

Figure 6 presents the cumulative distribution functions (CDFs) of average QoE for all sessions and algorithms on the training set, along with pairwise comparisons of the QoE components—bitrate, smoothness, and stall time. Figure 7 shows the corresponding results on the testing set. As shown, our proposed PA-MoE achieves state-of-the-art performance, even relative to the meta-learning method Merina, which leverages prior knowledge. These results underscore the substantial potential of optimizing adaptive bitrate (ABR) algorithms through the lens of plasticity.

• ***Comment 4*: *Figure 8 is difficult to interpret due to the lack of describing system changes. Similar to Figure 1, the relationship between the system environment and bitrate requires further explanation.***

Response: Thank you for the valuable comment.

We have added a description of the system-change mode and specified the expected bitrate response under environmental changes. "D" denotes documentaries, "L" live streams, and "N" news. The parameter coefficients for each video type are the same as those in Figure 1. We have revised the **Section V-F: Influence on System Internal States** subsection and added the following explanations (highlighted in blue in the manuscript) to further clarify our experimental results.

In this section, we present visualizations of the system's internal state metrics to analyze the behavior of different algorithms. This will help us gain a better understanding of their impact on overall QoE. The system environment

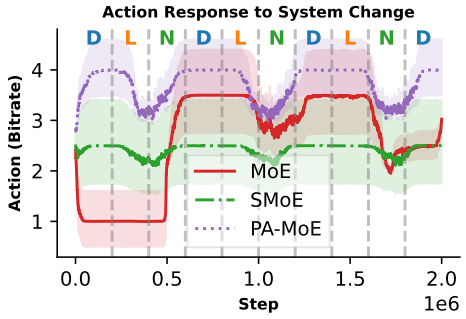

Figure 12: Illustration of Internal State Metrics Variations in a Video Streaming System under Shift QoE Conditions.

changes in Figure 12 are consistent with those in Figure 1. 'D' denotes documentaries, 'L' live streams, and 'N' news. The parameter coefficients for each video type are also the same as in Figure 1. Because the objective function changes abruptly, we expect the algorithm to adjust its behavior when the QoE coefficients vary. For example, when the objective function shifts from L to N, meaning that the bitrate coefficient in the QoE changes from 1 to β while the rebuffering coefficient changes from β to 1, the agent should correspondingly modify its action—specifically, by increasing the selected bitrate—to obtain a higher QoE reward.

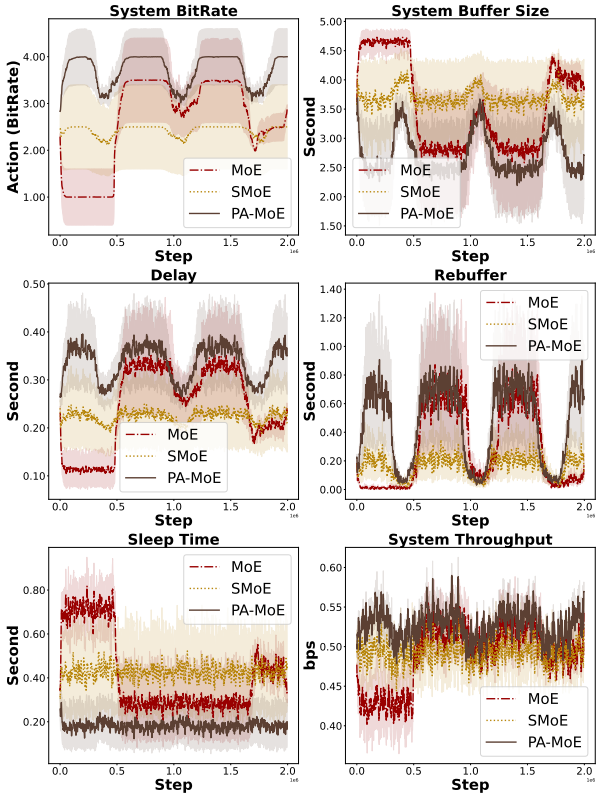

Figure 13: Variations in the Internal State Metrics of the Video Streaming System.

As shown in Figure 12, we can see that PA-MoE exhibits periodic variations in the action space, which correspond

to system changes. On the other hand, SMoE only shows minor fluctuations and fails to adapt quickly. In contrast, MoE does not demonstrate any periodic adaptation. Since the QoE coefficients change directly, using the QoE value itself to study plasticity is not appropriate. If the policy does not adapt its behavior in response to environmental changes, the direct modification of QoE coefficients will still cause the QoE value to fluctuate accordingly. Therefore, examining whether the actions exhibit periodic adjustments provides a more reliable reflection of the system's adaptive state.

Furthermore, for an AVS system, when the coefficient corresponding to the bitrate term in the QoE increases, the algorithm should select a higher bitrate to achieve a better QoE score. However, a higher bitrate requires more time to download, leading to increased delay and rebuffering time, while the buffer size decreases and the throughput increases. This represents a preliminary analysis of the internal state dynamics of the system.

Figure 13 illustrates detailed variations across the entire internal state of the system, revealing that PA-MoE more effectively utilizes the buffer size, resulting in higher throughput. From the perspective of internal system state variables, the performance improvement of PA-MOE mainly comes from reduced sleep time and more efficient buffer utilization. By adapting to changes in the action space, it can actively select actions that yield higher rewards. In contrast, MoE suffers from a loss of plasticity and fails to keep up with the rapidly changing system. This further validates the effectiveness of our proposed plasticity injection method.

• ***Comment 5: What is the decision delay of PA-MoE without GPU acceleration in deployment? Is this overhead reasonable for client-side implementation? A solid comparison of computational overhead with existing methods would further strengthen the superiority of PA-MoE.***

Response: We sincerely thank the reviewer for this thoughtful question regarding the decision delay and deployment feasibility of PA-MoE.

What is the decision delay of PA-MoE without GPU acceleration in deployment? Is this overhead reasonable for client-side implementation?

We evaluated PA-MoE's algorithmic latency without GPU acceleration. Over 14,400 decision inferences, the average decision delay was 0.8056 ms. Given that each video segment is 4 seconds long, a 0.8056 ms decision delay is well within an acceptable range.

The total number of parameters in Pensieve is 205,703, with a memory footprint of 0.78 MB. Merina has 208,775 parameters (0.80 MB), and PA-MoE has 821,651 parameters (3.13 MB). For modern computing platforms, these footprints are well within practical limits; moreover, model quantization and pruning can further reduce the memory requirements.

A solid comparison of computational overhead with existing methods would further strengthen the superiority of PA-MoE.

We evaluated 14,400 inference runs and measured the decision latency for the Buffer-based, Rate-based, RobustMPC, Pensieve, Merina, and PA-MoE, reporting the mean value, 50th, and 95th percentiles. The results are shown in Table 3.

Table 3: Inference time per decision (CPU-only, batch size = 1). Lower is better.

Method	Mean (ms)	p50 (ms)	p95 (ms)
Buffer-based	0.0026	0.0031	0.0032
Rate-based	0.0089	0.0087	0.0098
RobustMPC	0.1075	0.0696	0.3245
Pensieve	0.2984	0.2925	0.3266
Merina	0.4118	0.3947	0.4765
PA-MoE	0.8056	0.7994	0.8302

PA-MoE exhibits a decision latency nearly twice as high as that of existing learning-based methods. Notably, this latency does not scale with the number of experts; the incremental overhead is dominated by the router's feature processing and intra-MoE computation, rather than by the expert count itself.

As clarified in the revised manuscript (see **Section IV: Architectural Paradigm of the MoE Framework** subsection): "Although the architecture includes N experts, only one expert is selected for each input, keeping the per-sample complexity at $O(1)$."

This means that, during both training and inference, only a single expert (Top-1) is activated, while the others remain inactive, ensuring that the computational cost grows linearly with the number of active experts (=1) rather than with the total number of experts.

References

[1] H. Riiser, P. Vigmostad, C. Griwodz, and P. Halvorsen, "Commute path bandwidth traces from 3g networks: Analysis and applications," in *Proceedings of the 4th ACM Multimedia Systems Conference*, 2013, pp. 114–118.

[2] N. Kan, Y. Jiang, C. Li, W. Dai, J. Zou, and H. Xiong, "Improving generalization for neural adaptive video streaming via meta reinforcement learning," in *Proceedings of the 30th ACM International Conference on Multimedia*, 2022, pp. 3006–3016.

[3] F. Y. Yan, H. Ayers, C. Zhu, S. Fouladi, J. Hong, K. Zhang, P. Levis, and K. Winstein, "Learning in situ: a randomized experiment in video streaming," in *17th USENIX Symposium on Networked Systems Design and Implementation (NSDI 20)*, 2020, pp. 495–511.

[4] H. Mao, R. Netravali, and M. Alizadeh, "Neural adaptive video streaming with pensieve," in *Proceedings of the conference of the ACM special interest group on data communication*, 2017, pp. 197–210.

[5] T.-Y. Huang, R. Johari, N. McKeown, M. Trunnell, and M. Watson, "A buffer-based approach to rate adaptation: Evidence from a large video streaming service," in *Proceedings of the 2014 ACM conference on SIGCOMM*, 2014, pp. 187–198.

[6] J. Jiang, V. Sekar, and H. Zhang, "Improving fairness, efficiency, and stability in http-based adaptive video streaming with festive," in *Proceedings of the 8th international conference on Emerging networking experiments and technologies*, 2012, pp. 97–108.

[7] X. Yin, A. Jindal, V. Sekar, and B. Sinopoli, "A control-theoretic approach for dynamic adaptive video streaming over http," in *Proceedings of the 2015 ACM conference on special interest group on data communication*, 2015, pp. 325–338.

[8] M. Taha and A. Ali, "Smart algorithm in wireless networks for video streaming based on adaptive quantization," *Concurrency and Computation: Practice and Experience*, vol. 35, no. 9, p. e7633, 2023.

[9] M. Taha, A. Canovas, J. Lloret, and A. Ali, "A qoe adaptive management system for high definition video streaming over wireless networks," *Telecommunication Systems*, vol. 77, no. 1, pp. 63–81, 2021.

[10] M. Taha, A. Ali, J. Lloret, P. R. Gondim, and A. Canovas, "An automated model for the assessment of qoe of adaptive video streaming over wireless networks," *Multimedia Tools and Applications*, vol. 80, no. 17, pp. 26 833–26 854, 2021.

[11] A. Agarwal, S. M. Kakade, J. D. Lee, and G. Mahajan, "On the theory of policy gradient methods: Optimality, approximation, and distribution shift," *Journal of Machine Learning Research*, vol. 22, no. 98, pp. 1–76, 2021.

[12] A. Abels, D. Roijers, T. Lenaerts, A. Nowé, and D. Steckelmacher, "Dynamic weights in multi-objective deep reinforcement learning," in *International conference on machine learning*. PMLR, 2019, pp. 11–20.

[13] L. Zhang, G. Gao, and H. Zhang, "Towards data-efficient continuous learning for edge video analytics via smart caching," in *Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems*, 2022, pp. 1136–1140.

[14] D. Wu, P. Wu, M. Zhang, and F. Wang, "Mansy: Generalizing neural adaptive immersive video streaming with ensemble and representation learning," *IEEE Transactions on Mobile Computing*, 2024.

[15] G. Zhang, Z. Wang, H. Wei, M. Xiao, H. Yuan, D. Yu, and X. Cheng, "A novel spatial-temporal learning method for enhancing generalization in adaptive video streaming," *IEEE Transactions on Mobile Computing*, 2025.

[16] N. Kan, C. Li, Y. Jiang, W. Dai, J. Zou, H. Xiong, and L. Toni, "Merina+: Improving generalization for neural video adaptation via information-theoretic meta-reinforcement learning," *IEEE Transactions on Circuits and Systems for Video Technology*, 2025.

[17] S. Wang, J. Lin, and Y. Dai, "Mmvs: Enabling robust adaptive video streaming for wildly fluctuating and heterogeneous networks," *IEEE Transactions on Multimedia*, 2024.

[18] W. Li, X. Li, Y. Xu, Y. Yang, and S. Lu, "Metaabr: A meta-learning approach on adaptative bitrate selection for video streaming," *IEEE Transactions on Mobile Computing*, vol. 23, no. 3, pp. 2422–2437, 2023.

[19] W. Cai, J. Jiang, F. Wang, J. Tang, S. Kim, and J. Huang, "A survey on mixture of experts," *arXiv preprint arXiv:2407.06204*, 2024.

[20] J. Li, Z. Sun, X. He, L. Zeng, Y. Lin, E. Li, B. Zheng, R. Zhao, and X. Chen, "Locmoe: A low-overhead moe for large language model training." in *IJCAI*. ijcai.org, 2024, pp. 6377–6387.

[21] B. Lin, Z. Tang, Y. Ye, J. Cui, B. Zhu, P. Jin, J. Zhang, M. Ning, and L. Yuan, "Moe-llava: Mixture of experts for large vision-language models." *CoRR*, vol. abs/2401.15947, 2024.

[22] F. Xue, Z. Zheng, Y. Fu, J. Ni, Z. Zheng, W. Zhou, and Y. You, "Openmoe: an early effort on open mixture-of-experts language models," in *Proceedings of the 41st International Conference on Machine Learning*, ser. ICML'24. JMLR.org, 2024.

[23] H. Li, S. Lin, L. Duan, Y. Liang, and N. Shroff, "Theory on mixture-of-experts in continual learning," in *The Thirteenth International Conference on Learning Representations*, 2025. [Online]. Available: https://openreview.net/forum?id=7XgKAabsPp