Response to the Editors and Reviewers

Paper ID: MM-025164.R1

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 comments and those of reviewers, and to improve the quality of our manuscript. We are submitting a revised version of our manuscript and would like to thank you and the reviewers for your constructive comments.

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 sections of this response 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, which have helped improve our research.

Yours sincerely,

Zhiqiang He and Zhi Liu.

1. Response to the comments from Associate Editor

• ***Comment 1: Based on the enclosed set of reviews (**See note below about attachments), I have decided that the manuscript be ACCEPTED FOR PUBLICATION WITH MANDATORY MINOR REVISIONS INCLUDING ENGLISH USAGE (AQE). There is only one minor point that Reviewer 3 would like you to address. After that, the paper is ready for publication.***

Response:

We are grateful for the time and attention you devoted to our work. Regarding Reviewer 3's concerns about the statistical methods and the differences in effect across pairwise comparisons in the subfigures of Figs. 8 and 9, we have explained how the metric data were obtained and how the calculations were performed. **We provide explanations from four perspectives: the metric definition, the metric computation method, the tabulated numerical comparisons, and the open-source code link.**

Specifically, we have added the following content (in blue text):

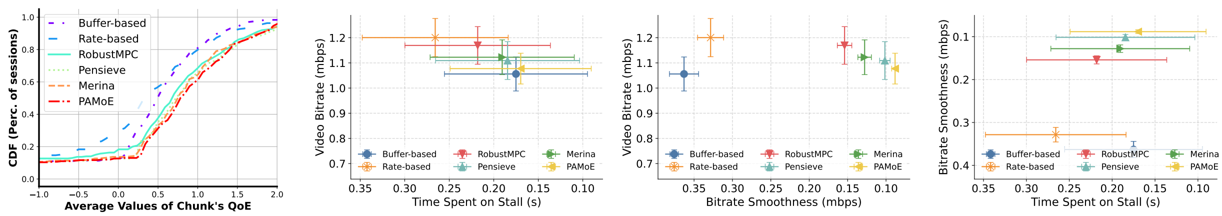

Figure 1: Comparing PA-MOE with recent ABR algorithms over the Train set.

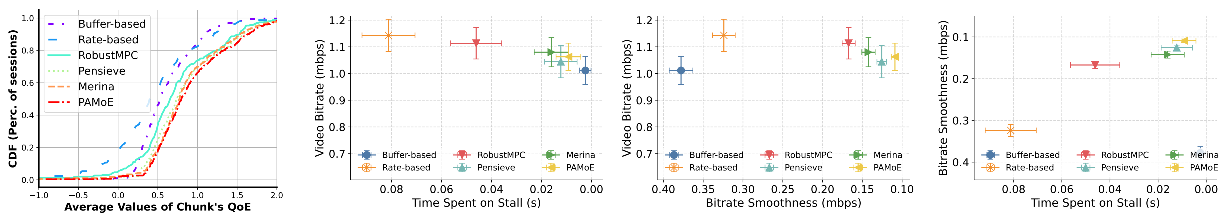

Figure 2: Comparing PA-MOE with recent ABR algorithms over the Test set.

In this subsection, we compare our approach with methods that rely on prior knowledge, including learning-based approaches such as Pensieve [1] and the meta-learning method Merina [2], as well as non-learning-based approaches such as RobustMPC, RateBased, and BufferBased [3]. To ensure a fair and credible comparison, we use the same QoE-component coefficients and the same network architecture as in Merina [4] for each expert in the MoE. Similar to Pensieve [1] and the meta-learning method Merina [2], we use the final trained model for performance evaluation. The network bandwidth is randomly sampled from either the training traces or the test traces, and each video download constitutes one episode.

For each algorithm, we run 300 episodes under identical settings and compute the average QoE for each episode; these per-episode averages are then used for comparing QoE performance across algorithms. The cumulative distribution functions (CDFs) plot the empirical cumulative distribution of QoE values across the 300 episodes. We aggregate the per-episode QoE values, compute a histogram using 500 equally spaced bins over the QoE range, take the cumulative sum of bin counts, and normalize by the total number of episodes to obtain the empirical CDF. Thus, at any position on the horizontal axis, the curve reports the fraction of sessions whose metric value is less than or equal to the corresponding horizontal value. No additional smoothing or parametric assumptions are used; the curves directly reflect the empirical distribution of the metric for each scheme. We perform pairwise comparisons of the QoE components—Bitrate Reward, Stall Time, and Smoothness—and compute the sample means together with 95% confidence intervals based on the Student's t-distribution.

Table 1: QoE and its components for different ABR algorithms on test dataset (mean over episodes).

Algorithm	Mean QoE ↑	Bitrate (Mbps) ↑	Rebuffer Time (s) ↓	Smoothness (Mbps) ↓
Rule-based methods				
Buffer Based	0.623	1.011 ± 0.053	0.002 ± 0.002	0.378 ± 0.015
Rate Based	0.470	1.143 ± 0.060	0.081 ± 0.011	0.324 ± 0.014
Robust MPC	0.748	1.113 ± 0.059	0.046 ± 0.010	0.167 ± 0.008
Learning-based methods				
Pensieve	0.866	1.044 ± 0.060	0.012 ± 0.007	0.126 ± 0.006
Merina	0.869	1.080 ± 0.055	0.016 ± 0.007	0.142 ± 0.084
PA-MoE	**0.914**	1.063 ± 0.051	0.009 ± 0.005	0.109 ± 0.004

Note: ↑ indicates larger is better; ↓ indicates smaller is better. Bitrate, Rebuffer Time, and Smoothness are reported as mean ± 95% Confidence Interval (CI); Mean QoE is reported as mean only.

Figure 1 presents the 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 2 shows the corresponding results on the testing set. Table 1 reports the QoE and its components achieved by different ABR algorithms on the test 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.

2. Response to Reviewer 3

2.1. Comments

The authors have addressed most of my concerns effectively. The only remaining point is to clarify the statistical methods and effect differences for the pairwise comparisons in the subfigures of Fig. 8 and 9.

Response:

Thank you for your positive feedback. We sincerely appreciate your patience in carefully reading our manuscript and for providing valuable suggestions.

Regarding the statistical methods you mentioned, we have added a detailed description of the metric definition and the metric computation procedure. To better characterize the performance gaps in our pairwise comparisons, we have also added a table reporting the exact results on the test set. Finally, we will release the source code, which includes the full implementation details.

We first introduce how the QoE metric is computed.

1. **Episode definition:** In our setup, each episode corresponds to downloading a single video.
2. **Collection of QoE data:** For learning-based methods, to ensure a fair comparison with prior work, we follow Pensieve [1] and the meta-learning method Merina [2] by using the converged model for forward inference to obtain the inferred QoE value for each episode.
3. **QoE computation for plotting:** The same video is simulated and transmitted 300 times under different network bandwidth conditions (bandwidth traces). After each transmission, we compute an average QoE for that episode, and these per-episode average QoE values are used for subsequent plotting.

When the simulations are conducted on the training bandwidth traces, the corresponding figure is **Figure 3. Comparing PA-MOE with recent ABR algorithms over the Train set.** When the simulations are conducted on the testing bandwidth traces, the corresponding figure is **Figure 4. Comparing PA-MOE with recent ABR algorithms over the Test set.**

We then explain how the figures are generated.

CDF Plot Generation: From the 300 simulation runs, we obtain 300 average QoE values. Since these QoE values vary across runs, we construct an empirical distribution by taking the QoE range (from the minimum to the maximum) and dividing it into 500 equal-width intervals. We then build a probability histogram by counting how many QoE values fall into each interval, which yields the empirical distribution of QoE. Based on this distribution, we compute and plot the empirical cumulative distribution function (CDF).

Plot Pairwise Comparisons: In addition, using the same set of 300 per-episode average QoE values, we perform pairwise comparisons among the QoE component metrics and compute confidence intervals, producing three subplots: Video Bitrate (Mbps) vs. Time Spent on Stall (s), Video Bitrate (Mbps) vs. Bitrate Smoothness (Mbps), and Bitrate Smoothness (Mbps) vs. Time Spent on Stall (s).

Effect differences for the pairwise comparisons: We added Table 2 to present the exact numerical values from the pairwise comparisons, which more accurately reflect the performance differences.

Specific source-code implementation: More detailed information can be found in the code, and we have included the code link in the Abstract section. The code is available at https://github.com/tinyzqh/PA-MoE. The code will be released publicly once it has been properly organized and finalized.

Specifically, we have added the following content (highlighted in blue):

In this subsection, we compare our approach with methods that rely on prior knowledge, including learning-based approaches such as Pensieve [1] and the meta-learning method Merina [2], as well as non-learning-based approaches such as RobustMPC, RateBased, and BufferBased [3]. To ensure a fair and credible comparison, we use the same QoE-component coefficients and the same network architecture as in Merina [4] for each expert in the MoE. Similar to Pensieve [1] and the meta-learning method Merina [2], we use the final trained model for performance evaluation. The network bandwidth is randomly sampled from either the training traces or the test traces, and each video download constitutes one episode.

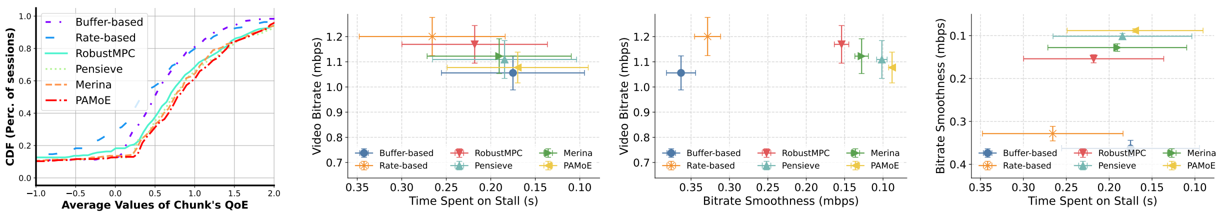

Figure 3: Comparing PA-MOE with recent ABR algorithms over the Train set.

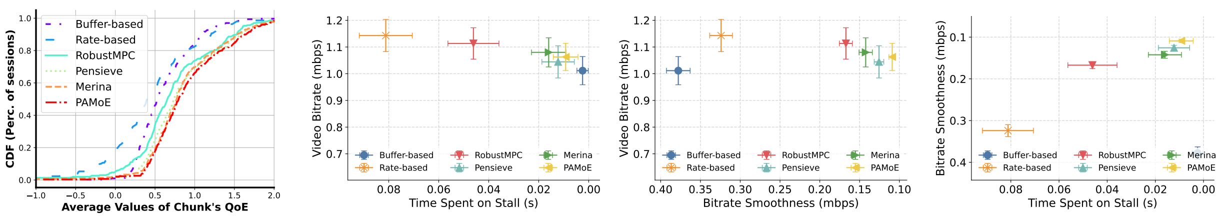

Figure 4: Comparing PA-MOE with recent ABR algorithms over the Test set.

For each algorithm, we run 300 episodes under identical settings and compute the average QoE for each episode; these per-episode averages are then used for comparing QoE performance across algorithms. The cumulative distribution functions (CDFs) plot the empirical cumulative distribution of QoE values across the 300 episodes. We aggregate the per-episode QoE values, compute a histogram using 500 equally spaced bins over the QoE range, take the cumulative sum of bin counts, and normalize by the total number of episodes to obtain the empirical CDF. Thus, at any position on the horizontal axis, the curve reports the fraction of sessions whose metric value is less than or equal to the corresponding horizontal value. No additional smoothing or parametric assumptions are used; the curves directly reflect the empirical distribution of the metric for each scheme. We perform pairwise comparisons of the QoE components—Bitrate Reward, Stall Time, and Smoothness—and compute the sample means together with 95% confidence intervals based on the Student's t-distribution.

Table 2: QoE and its components for different ABR algorithms on test dataset (mean over episodes).

Algorithm	Mean QoE ↑	Bitrate (Mbps) ↑	Rebuffer Time (s) ↓	Smoothness (Mbps) ↓
Rule-based methods				
Buffer Based	0.623	1.011 ± 0.053	0.002 ± 0.002	0.378 ± 0.015
Rate Based	0.470	1.143 ± 0.060	0.081 ± 0.011	0.324 ± 0.014
Robust MPC	0.748	1.113 ± 0.059	0.046 ± 0.010	0.167 ± 0.008
Learning-based methods				
Pensieve	0.866	1.044 ± 0.060	0.012 ± 0.007	0.126 ± 0.006
Merina	0.869	1.080 ± 0.055	0.016 ± 0.007	0.142 ± 0.084
PA-MoE	**0.914**	1.063 ± 0.051	0.009 ± 0.005	0.109 ± 0.004

Note: ↑ indicates larger is better; ↓ indicates smaller is better. Bitrate, Rebuffer Time, and Smoothness are reported as mean ± 95% Confidence Interval (CI); Mean QoE is reported as mean only.

Figure 3 presents the 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 4 shows the corresponding results on the testing set. Table 2 reports the QoE and its components achieved by different ABR algorithms on the test 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.

In closing, we sincerely appreciate your careful evaluation and valuable feedback. Your comments have been instrumental in improving the clarity and overall quality of our paper.

References

[1] 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.

[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] 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.

[4] 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.