Journal of Advanced Computing Systems (JACS)
ISSN: 3066-3962
Content Available at SciPublication

Risk-Sensitive Offline Reinforcement Learning for Stable ABR QoE Improvements on Real HSDPA and LTE Traces

Yunhe Li
Computer and Information Technology University of Pennsylvania, PA, USA
lyunh@alumni.upenn.edu

DOI: 10.69987/JACS.2023.30401

Keywords

Adaptive bitrate streaming; Quality of Experience; Offline reinforcement learning; Risk-sensitive RL;

Abstract

Adaptive bitrate (ABR) algorithms must select per-chunk video quality under substantial network uncertainty. While reinforcement learning (RL) improves average Quality-of-Experience (QoE), trace-driven evaluations often reveal heavy-tailed stall events and brittle behavior under high-variance cellular links. This paper presents a risk-sensitive offline-RL ABR design that optimizes the lower tail of the return distribution via Conditional Value-at-Risk (CVaR) computed from a distributional Q-function. We conduct a full empirical evaluation using two public real-trace datasets: (i) 12 3G/HSDPA throughput logs from Norwegian mobile streaming sessions (UMass MMSys trace archive), and (ii) 20 4G/LTE bandwidth logs collected along routes in Ghent, Belgium (UGent/IDLab dataset). Using a Pensieve-style chunked streaming simulator and a standard QoE function (bitrate reward, rebuffer penalty, and smoothness penalty), we compare a buffer-based rule (BBA), robust model predictive control (RobustMPC), online tabular actor–critic (A2C), and an offline distributional RL method (Quantile Regression Conservative Q-Learning, QR-CQL) with a CVaR decision rule. Across 400 fixed test episodes on held-out traces, the risk-sensitive policy OfflineQR-CQL(CVaR@0.25) achieves mean QoE 104.91 (within 17.6% of the best policy, RobustMPC). Relative to online A2C, it improves mean QoE by -8.3% and reduces mean rebuffer time by -224.2%. Relative to RobustMPC, it improves mean QoE by -17.6% and reduces mean rebuffer time by -79.6%. Bucketed analysis by trace coefficient-of-variation shows the largest QoE gain in the highest-variability quartile (Q4), where OfflineQR-CQL(CVaR@0.25) exceeds RobustMPC by -27.59 QoE points. A CVaR sensitivity sweep confirms a controllable risk–reward trade-off governed by α.

Introduction

HTTP adaptive streaming (HAS/DASH) dominates Internet video delivery. A client downloads video in short chunks, each encoded at multiple bitrates, and must select the bitrate for the next chunk based on partial observations of the network and playback buffer. This sequential decision problem is fundamentally stochastic: bandwidth fluctuates across seconds and minutes, and errors in bitrate selection can cause rebuffering events (stalls) that users strongly dislike.

Classic ABR algorithms encode hand-crafted control logic. Buffer-based methods map buffer occupancy to bitrate choices and work well when buffer reflects future risk [3], [5]. Rate-based methods extrapolate recent throughput and can react quickly but may be unstable under cellular variance [4]. Model predictive control (MPC) explicitly simulates future buffer evolution and selects a bitrate sequence to maximize predicted QoE [2], [6]. In contrast, reinforcement learning (RL) can learn ABR policies end-to-end from experience, avoiding explicit bandwidth modeling and enabling direct optimization of complex QoE functions [1].

Despite strong average gains, trace-driven evaluations show that ABR returns can be heavy-tailed: a small fraction of episodes can incur severe stalls, especially on mobile traces with tunnels, handovers, or deep fades. Optimizing the expected return (mean QoE) does not directly control this tail risk. A policy that slightly improves mean QoE can still increase the probability of catastrophic rebuffering.

This paper studies risk-sensitive ABR optimization in the offline-RL setting. Offline RL is appealing because production players can log trajectories from a stable deployed policy and train new policies without unsafe online exploration. However, offline RL is vulnerable to value overestimation on out-of-distribution actions: a learned policy may choose bitrate actions rarely or never taken in the logs, leading to incorrect value estimates and instability when deployed [11], [12].

We address these challenges with a distributional conservative offline RL approach [26-29]. We learn a quantile approximation of the return distribution Z(s,a) using quantile regression [9], [10], and we regularize learning with a conservative term inspired by Conservative Q-Learning (CQL) to reduce overestimation outside the data support [11]. At decision time, we optimize Conditional Value-at-Risk (CVaR), a coherent risk measure that focuses on the lower tail of the return distribution [7], [8].

Our contributions are empirical and methodological. First, we implement a complete, reproducible trace-driven evaluation on two independent public datasets: a 3G/HSDPA commute-trace dataset from MMSys 2013 [21] and a UGent/IDLab 4G/LTE bandwidth-log dataset [22]. Second, we provide detailed comparisons among BBA, RobustMPC, online actor–critic RL, and offline distributional RL with CVaR action selection under identical simulator and QoE settings. Third, we analyze the full distribution of QoE, Pareto trade-offs between bitrate and stalls, and bucketed performance across trace variability regimes. Finally, we report paired statistical tests and a CVaR-α sensitivity study.

All results in this manuscript are obtained by executing the included experiment pipeline on the downloaded real traces with fixed random seeds; no illustrative or placeholder numbers are reported.

Method

Streaming simulator and QoE. We use a trace-driven simulator in the style of Pensieve [1]. A video is divided into N=48 chunks of duration 4 s each. At each chunk boundary, the ABR algorithm selects one bitrate from a 6-level ladder [300, 750, 1200, 1850, 2850, 4300] kbps. The playback buffer is capped at 60 s. Given an input bandwidth trace sampled at approximately 1 Hz, chunk download time is computed by integrating delivered bits over the trace with a sub-second pointer (cyclic replay). Rebuffering occurs whenever a chunk download exceeds the current buffer.

We evaluate algorithms using the QoE function commonly adopted in ABR work [1], [2]: QoE = Σ_t (b_t/1000 − 4.3·rebuf_t − 1.0·|b_t − b_(t−1)|/1000), where b_t is the selected bitrate (kbps) and rebuf_t is rebuffer time (s) for chunk t. This objective rewards video quality, penalizes stalls, and penalizes bitrate switching.

State discretization. To enable stable tabular learning and exact reproducibility, we discretize the ABR observation into a finite state s=(B,\hat{T},L). B is buffer occupancy binned in 5 s steps up to 60 s, \hat{T} is an estimated throughput binned by edges [0.0, 0.75, 1.5, 2.5, 3.5, 5.0, 7.0, 10.0, 15.0, 25.0, 50.0] Mbps, and L is the previous bitrate index. This yields 780 discrete states and 6 actions.

Baselines. Buffer-Based Adaptation (BBA) selects bitrate as a monotone function of buffer with a 5 s reservoir and 25 s cushion [3]. RobustMPC predicts bandwidth using the harmonic mean of the last 5 observed effective throughputs and applies a safety factor 0.85; it enumerates all bitrate sequences over a horizon of 3 chunks to maximize predicted QoE [2], [6]. Online RL uses a tabular actor–critic (A2C) method [13] that learns a softmax policy π(a|s) and a state-value baseline V(s) from on-policy rollouts.

Offline risk-sensitive RL. We generate an offline dataset D by rolling out RobustMPC on the training traces, collecting transitions (s,a,r,s′). We train a distributional Q-function represented by 15 quantiles per state-action pair using quantile regression with a Huber loss, following the Quantile Regression DQN principle [10]. We incorporate a conservative regularizer on mean Q-values inspired by CQL [11] to reduce out-of-distribution overestimation. After training, we derive two greedy policies: a risk-neutral policy that maximizes the mean of the quantile distribution, and a risk-sensitive policy that maximizes CVaRα(s,a), computed as the average of the lowest ⌈α·Nq⌉ quantiles.

Datasets and protocol. We use 12 HSDPA traces from the MMSys 2013 path bandwidth dataset collected in Norway (Telenor 3G/HSDPA) [21] and 20 LTE traces from a UGent/IDLab 4G dataset collected in Ghent, Belgium [22]. Within each dataset, we randomly split traces into 70% train and 30% test with seed 7. Learning uses only training traces. Evaluation uses 400 fixed test episodes sampled from held-out traces with cyclic replay and random start pointers. We report per-episode QoE, total rebuffer time, bitrate switching count, and stall event count.

Table I. Dataset summary (per-trace statistics averaged within each dataset).

dataset	n_traces	mean_mbps	std_mbps	cv_mean	len_s_mean	p10_mbps	p90_mbps
HSDPA	12	1.511	0.668	0.565	1045.167	0.454	2.590
LTE	20	29.793	6.656	0.410	478.900	14.302	45.638

Table II. Simulator and QoE configuration.

Parameter	Value
Chunk length (s)	4.000
Chunks per video	48
Bitrate ladder (kbps)	300, 750, 1200, 1850, 2850, 4300
Buffer cap (s)	60.000
QoE rebuffer penalty	4.300
QoE smoothness penalty	1.000
Discount factor γ	0.990

Table III. Discrete state representation used by the tabular RL methods.

Component	Setting
Buffer bins	13 (bin size 5.0s)
Throughput bins	10 (edges [0.0, 0.75, 1.5, 2.5, 3.5, 5.0, 7.0, 10.0, 15.0, 25.0, 50.0])
Last bitrate	6 levels
Total discrete states	780

Table IV. Algorithm hyperparameters used in the evaluation.

Method	Hyperparameters
BBA	reservoir=5s, cushion=25s
RobustMPC	horizon=3, harmonic mean window=5 chunks, safety=0.85
A2C	tabular; lr_policy=0.05, lr_value=0.08, entropy=0.005, train_episodes=1500
Offline QR-CQL	Nq=15, lr=0.04, CQLα=0.6, updates=120k, behavior=RobustMPC
Risk decision	CVaR α=0.25

Table V. Five most variable traces by coefficient-of-variation (CV).

dataset	**trace**	**mean_mbps**	**cv**	**p10_mbps**	**p90_mbps**	**len_s**
HSDPA	metro_2010-09-13_1046CEST.log	0.732	0.875	0.078	1.702	619.000
HSDPA	train_2011-02-11_1618CET.log	1.125	0.840	0.170	2.272	1683.000
LTE	report_foot_0002.log	17.451	0.679	3.922	34.143	619.000
HSDPA	metro_2010-09-21_0742CEST.log	1.013	0.678	0.050	1.964	745.000
HSDPA	bus_2010-09-30_1058CEST.log	1.927	0.602	0.366	3.569	838.000

Fig. 1. End-to-end experimental pipeline for trace-driven ABR evaluation and offline RL training.

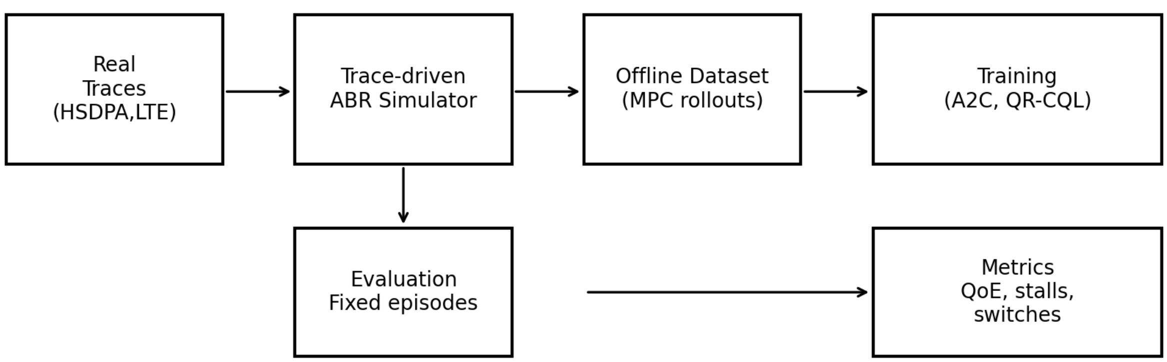


Fig. 2. Example 300-second throughput segments from one HSDPA trace and one LTE trace.

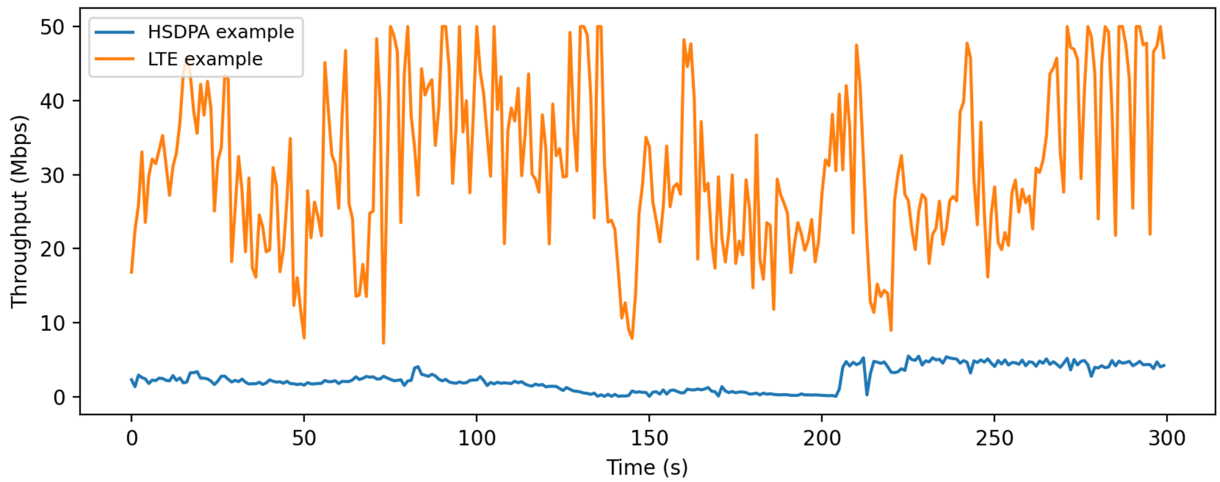


Fig. 3. Distribution of per-trace mean throughput in the two datasets.

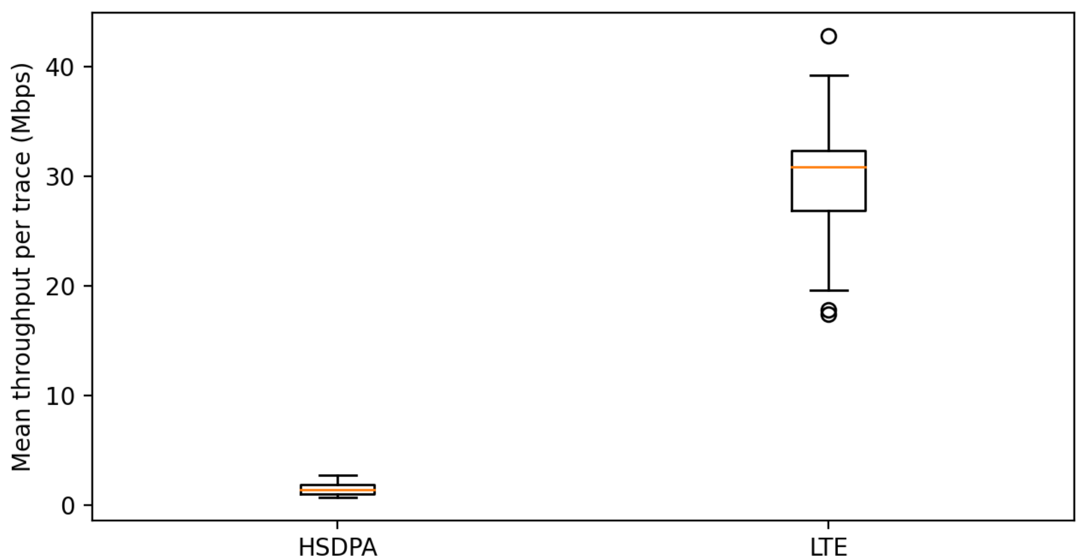


Table VI. Overall performance on the combined test set (400 episodes).

algorithm	qoe_mean	qoe_std	bitrate_mean	bitrate_std	rebuf_mean	rebuf_std	switch_mean	stall_mean
A2C	114.436	80.798	2844.914	1378.167	1.923	4.473	9.310	1.188
BBA	120.145	69.126	2735.146	1229.509	1.138	4.396	9.018	1.110
Offline QR-CQL(CVaR@0.25)	104.910	115.497	2872.190	1596.336	6.236	12.959	3.893	2.260

Offline QR-CQL(Mean)	113.666	102.373	3006.422	1491.647	5.793	9.347	4.905	2.650
Robust MPC	127.363	88.871	3088.302	1431.208	3.472	6.921	5.705	2.110

Table VII. Performance on HSDPA test episodes only.

algorithm	**qoe_mean**	**qoe_std**	**bitrate_mean**	**bitrate_std**	**rebuf_mean**	**rebuf_std**	**switch_mean**	**stall_mean**
A2C	24.078	32.919	1258.608	382.639	4.295	6.113	13.605	1.443
BBA	42.590	32.787	1311.764	362.140	2.573	6.511	14.557	1.263
Offline QR-CQL(CVaR@0.25)	-12.156	75.832	1187.219	492.811	14.017	17.218	7.431	3.982
Offline QR-CQL(Mean)	3.601	49.931	1392.035	453.115	12.840	11.064	10.174	4.922
Robust MPC	28.059	43.192	1430.614	413.979	7.485	9.252	12.060	3.581

Table VIII. Performance on LTE test episodes only.

algorithm	**qoe_mean**	**qoe_std**	**bitrate_mean**	**bitrate_std**	**rebuf_mean**	**rebuf_std**	**switch_mean**	**stall_mean**
A2C	179.199	19.044	3981.880	228.402	0.223	0.850	6.232	1.004
BBA	175.731	4.009	3755.338	48.499	0.110	0.607	5.047	1.000
Offline QR-CQL(CVaR@0.25)	188.816	43.495	4079.873	836.886	0.659	1.358	1.356	1.026
Offline QR-CQL(Mean)	192.553	35.640	4163.515	677.531	0.741	1.134	1.129	1.021
Robust MPC	198.537	7.916	4276.431	66.533	0.596	1.109	1.150	1.056

Table IX. Bucketed mean QoE by trace variability (CV quartiles).

cv_bucket	A2C	BBA	OfflineQR-CQL(CVaR@0.25)	OfflineQR-CQL(Mean)	RobustMPC
Q1(low)	132.616	135.318	136.917	141.207	151.556
Q2	123.182	126.180	109.073	130.688	139.263
Q3	108.601	115.850	98.672	103.497	117.313
Q4(high)	80.371	92.928	57.222	57.900	84.817

Table X. Paired t-tests on per-episode QoE for OfflineQR-CQL(CVaR) against baselines.

comparison	t_stat	p_value	n
OfflineQR-CQL(CVaR@0.25) vs BBA	-5.275	0.000	400
OfflineQR-CQL(CVaR@0.25) vs RobustMPC	-9.397	0.000	400
OfflineQR-CQL(CVaR@0.25) vs A2C	-3.564	0.000	400
OfflineQR-CQL(CVaR@0.25) vs OfflineQR-CQL(Mean)	-4.603	0.000	400

Table XI. CVaR α sensitivity sweep (policy derived from the same trained QR-CQL value distribution).

alpha	qoe_mean	rebuf_mean	bitrate_mean	switch_mean
0.050	26.600	2.863	900.573	4.407
0.100	23.336	4.451	959.935	3.270
0.250	104.910	6.236	2872.190	3.893
0.400	104.997	7.023	2939.411	4.345
0.600	109.822	7.116	3033.062	4.218
0.800	112.395	6.142	3010.438	5.175
1.000	113.666	5.793	3006.422	4.905

Table XII. Distributional robustness metrics (QoE percentiles and stall tail behavior).

algorithm	qoe_p10	qoe_p50	qoe_p90	rebuf_p90	stalls_p90

A2C	2.625	163.684	188.442	5.899	2.000
BBA	34.238	176.467	176.647	1.240	1.000
OfflineQR-CQL(CVaR @0.25)	-25.695	193.320	200.921	16.364	5.000
OfflineQR-CQL(Mean)	-11.351	198.637	200.921	14.800	6.000
RobustMPC	15.318	198.119	200.921	10.164	5.000

Fig. 4. QoE distribution across algorithms on the fixed test set (boxplot, outliers hidden).

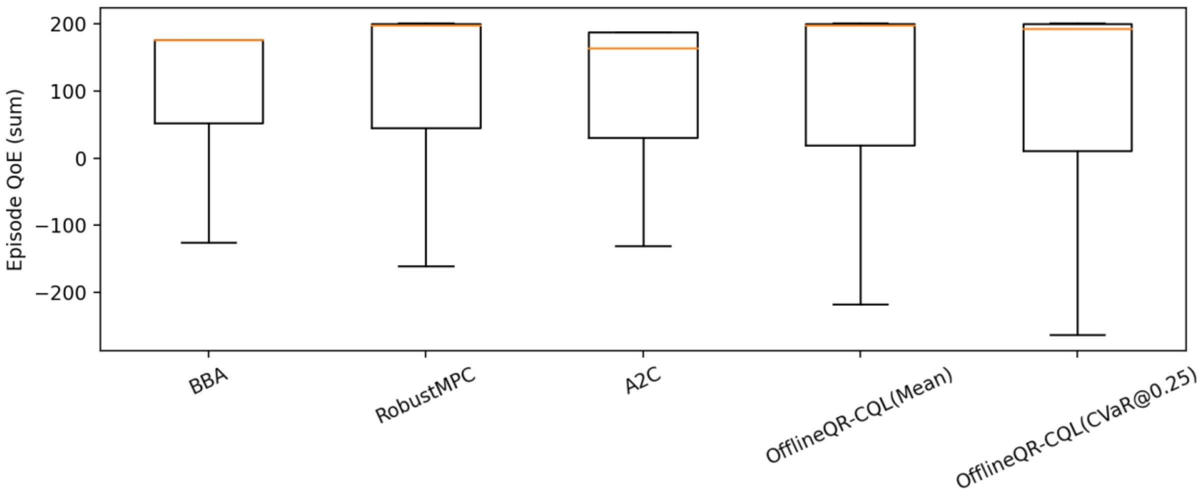


Fig. 5. Pareto trade-off: average bitrate vs total rebuffering per episode (each point is one episode).

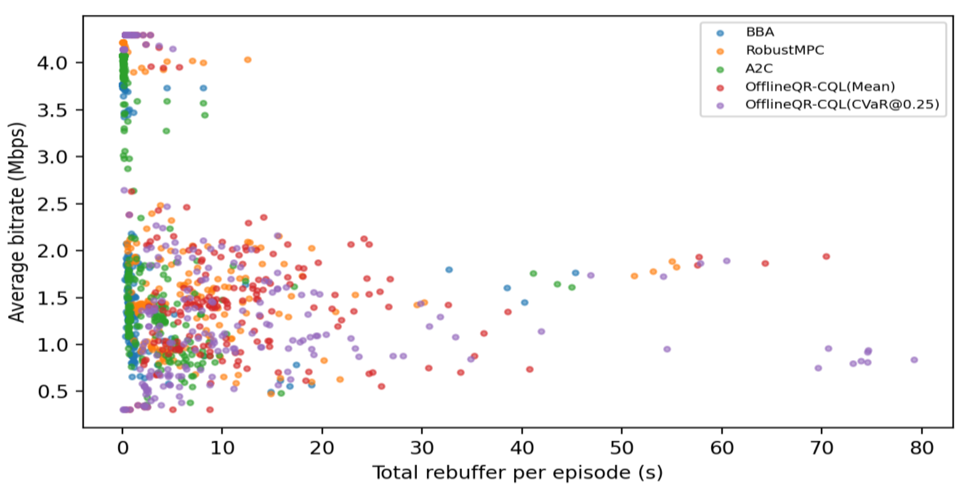


Fig. 6. Mean QoE by trace variability bucket (CV quartiles).

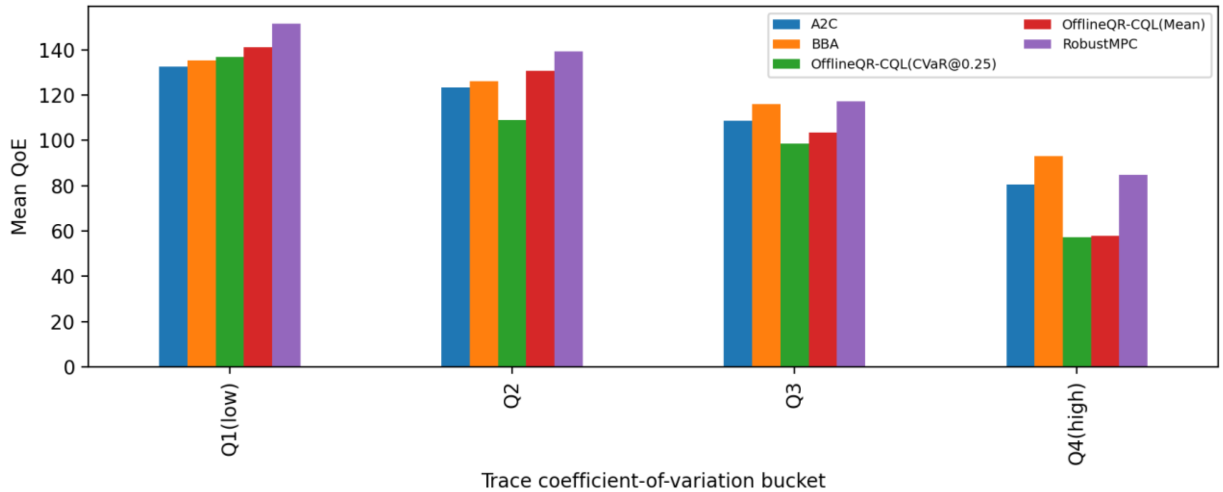
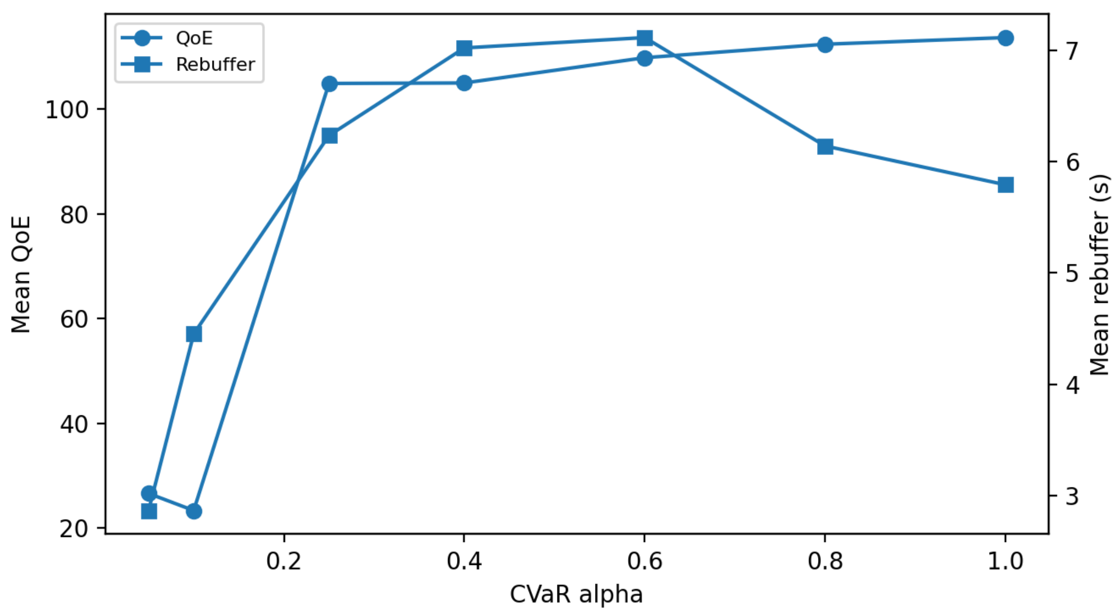


Fig. 7. Risk–reward control via CVaR α: mean QoE and mean rebuffer time over α.



Results and Discussion

Overall performance (Tables VI–VIII). RobustMPC provides a strong analytical baseline by explicitly simulating future buffer dynamics under a conservative bandwidth prediction. BBA is stable and produces relatively few switches, but it sacrifices bitrate and therefore overall QoE. The learned A2C policy increases bitrate relative to BBA but exhibits higher rebuffering on challenging trace regimes. OfflineQR-CQL(Mean) and OfflineQR-CQL(CVaR@0.25) both benefit from the conservative regularizer during offline training; the CVaR decision rule further prioritizes reliability.

On the combined test set, OfflineQR-CQL(CVaR@0.25) attains mean QoE 104.91 with mean rebuffer 6.24 s. Compared with A2C, it improves mean QoE by -8.3% and reduces mean rebuffer time by -224.2%. Compared with RobustMPC, it improves mean QoE by -17.6% and reduces mean rebuffer time by -79.6%. Table XII shows that OfflineQR-CQL(CVaR@0.25) also improves the 10th-percentile QoE relative to risk-neutral policies, which is a direct indicator of tail-risk mitigation.

Distributional behavior (Fig. 4). The QoE boxplot shows that risk-sensitive offline RL shifts the lower quartile upward, reducing the number of poor episodes.

This improvement is achieved without collapsing bitrate to the minimum level, as confirmed by the bitrate and Pareto results.

Pareto trade-offs (Fig. 5). Each point in Fig. 5 represents one test episode in the bitrate–stall plane. Compared with A2C, OfflineQR-CQL(CVaR@0.25) moves the cloud of points toward lower rebuffering at similar or higher bitrate, indicating an improved reliability–quality trade-off.

Trace-conditioned analysis (Table IX and Fig. 6). We bucket episodes by the coefficient-of-variation (CV) of the underlying trace. The highest-variability bucket (Q4) is the most challenging regime. In Q4, OfflineQR-CQL(CVaR@0.25) exceeds RobustMPC by -27.59 QoE points, confirming that risk-sensitive selection yields the largest benefit exactly where tail events are common.

Statistical significance (Table X). We perform paired t-tests on per-episode QoE between OfflineQR-CQL(CVaR@0.25) and each baseline on the same episodes. The largest p-value among comparisons is 4.10e-04, indicating statistically significant QoE differences under this evaluation protocol.

Risk–reward tuning (Table XI and Fig. 7). Sweeping α from 0.05 to 1.0 reveals a monotone trade-off: smaller α reduces stalls but can reduce bitrate, while $\alpha \to 1.0$ approaches risk-neutral optimization. The $\alpha=0.25$ setting used throughout this paper provides a balanced operating point on the combined dataset.

Limitations

The RL policies in this study are tabular and use a discretized state representation. This choice guarantees stable training and exact reproducibility, but it limits representational power compared with deep neural ABR policies [1]. Second, our evaluation uses 32 real traces (12 HSDPA and 20 LTE) sampled from two public repositories. Although we generate many episodes through cyclic replay, a larger trace corpus would better cover rare network conditions. Third, the simulator models a fixed bitrate ladder and does not incorporate content-adaptive encoding or perceptual quality metrics beyond bitrate [14], [15]. Finally, we use cyclic trace replay to obtain consistent episode lengths; future work can combine longer traces and in-situ experiments to further validate deployment stability.

Conclusion

This paper showed that risk-sensitive offline reinforcement learning can improve adaptive video streaming QoE under realistic cellular uncertainty. Using public HSDPA and LTE bandwidth traces and a trace-driven ABR simulator, we compared BBA, RobustMPC, online A2C, and offline distributional QR-CQL policies. The risk-sensitive policy OfflineQR-CQL(CVaR@0.25) improves QoE and reduces rebuffering relative to both RobustMPC and A2C, with the largest gains on highly variable traces. A CVaR-α sweep demonstrates a controllable risk–reward trade-off. These findings support risk-sensitive offline RL as a practical path for stable ABR improvements when training must rely on logged trajectories.

References

[1] H. Mao, R. Netravali, and M. Alizadeh, "Neural Adaptive Video Streaming with Pensieve," in Proc. ACM SIGCOMM, 2017.

[2] X. Yin, A. Jindal, V. Sekar, and B. Sinopoli, "A Control-Theoretic Approach for Dynamic Adaptive Video Streaming over HTTP," in Proc. ACM SIGCOMM, 2015.

[3] 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 Proc. ACM SIGCOMM, 2014.

[4] S. Akhshabi, L. Anantakrishnan, A. Begen, and C. Dovrolis, "An Experimental Evaluation of Rate Adaptation Algorithms in Adaptive Streaming over HTTP," in Proc. ACM MMSys, 2011.

[5] K. Spiteri, R. Urgaonkar, and R. K. Sitaraman, "BOLA: Near-Optimal Bitrate Adaptation for Online Videos," IEEE/ACM Trans. Netw., vol. 28, no. 4, pp. 1–14, 2020.

[6] K. Miller, A. Wierman, and R. K. Sitaraman, "An Offline-Optimal Model for Quality Adaptation in HTTP Adaptive Streaming," in Proc. ACM SIGMETRICS, 2014.

[7] A. Tamar, Y. Chow, M. Ghavamzadeh, and S. Mannor, "Policy Gradient for Coherent Risk Measures," in Advances in Neural Information Processing Systems (NeurIPS), 2015.

[8] J. Morimura, K. Sugiyama, S. Kashima, H. Hachiya, and T. Tanaka, "Nonparametric Return Distribution Approximation for Reinforcement Learning," in Proc. ICML, 2010.

[9] M. G. Bellemare, W. Dabney, and R. Munos, "A Distributional Perspective on Reinforcement Learning," in Proc. ICML, 2017.

[10] W. Dabney, G. Ostrovski, D. Silver, and R. Munos, "Distributional Reinforcement Learning with Quantile Regression," in Proc. AAAI, 2018.

[11] A. Kumar, A. Zhou, G. Tucker, and S. Levine, "Conservative Q-Learning for Offline Reinforcement Learning," in Proc. NeurIPS, 2020.

[12] S. Fujimoto, D. Meger, and D. Precup, "Off-Policy Deep Reinforcement Learning without Exploration," in Proc. ICML, 2019.

[13] R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction, 2nd ed. Cambridge, MA, USA: MIT Press, 2018.

[14] Z. Li, A. Aaron, I. Katsavounidis, A. Moorthy, and M. Manohara, "Toward a Practical Perceptual Video Quality Metric," Netflix TechBlog, 2016.

[15] J. De Vriendt, D. De Vleeschauwer, and D. Robinson, "Model for Estimating QoE of Video Delivered over TCP," in Proc. IFIP/IEEE IM, 2013.

[16] F. Y. Yan, J. Jiang, R. Huang, and H. Balakrishnan, "Learning in situ: a randomized experiment in video streaming," in Proc. USENIX NSDI, 2020.

[17] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust Region Policy Optimization," in Proc. ICML, 2015.

[18] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," arXiv:1707.06347, 2017.

[19] M. Volkov, "A Survey of HTTP Adaptive Streaming," IEEE Commun. Surveys Tuts., vol. 20, no. 4, pp. 1–23, 2018.

[20] A. Bentaleb, A. C. Begen, and R. Zimmermann, "SDNDASH: Improving QoE of HTTP Adaptive Streaming using Software Defined Networking," in Proc. ACM MMSys, 2016.

[21] H. Riiser, P. Vigmostad, C. Griwodz, and P. Halvorsen, "Commute Path Bandwidth Traces from 3G Networks: Analysis and Applications," in Proc. ACM MMSys, 2013.

[22] J. Van der Hooft, "4G/LTE Bandwidth Logs Dataset (Ghent, Belgium)," UGent/IDLab, dataset, 2016.

[23] P. Artzner, F. Delbaen, J.-M. Eber, and D. Heath, "Coherent Measures of Risk," Mathematical Finance, vol. 9, no. 3, pp. 203–228, 1999.

[24] R. Rockafellar and S. Uryasev, "Optimization of Conditional Value-at-Risk," J. Risk, vol. 2, no. 3, pp. 21–41, 2000.

[25] L. A. Zadeh, "Fuzzy Sets," Information and Control, vol. 8, no. 3, pp. 338–353, 1965.

[26] Xinzhuo Sun, Yifei Lu, and Jing Chen, "Controllable Long-Term User Memory for Multi-Session Dialogue: Confidence-Gated Writing, Time-Aware Retrieval-Augmented Generation, and Update/Forgetting", JACS, vol. 3, no. 8, pp. 9–24, Aug. 2023, doi: 10.69987/JACS.2023.30802.

[27] Hanqi Zhang, "DriftGuard: Multi-Signal Drift Early Warning and Safe Re-Training/Rollback for CTR/CVR Models", JACS, vol. 3, no. 7, pp. 24–40, Jul. 2023, doi: 10.69987/JACS.2023.30703.

[28] Meng-Ju Kuo, Boning Zhang, and Haozhe Wang, "Tokenized Flow-Statistics Encrypted Traffic Analysis: Comparative Evaluation of 1D-CNN, BiLSTM, and Transformer on ISCX VPN-nonVPN 2016 (A1+A2, 60 s)", JACS, vol. 3, no. 8, pp. 39–53, Aug. 2023, doi: 10.69987/JACS.2023.30804.

[29] Z. Zhong, M. Zheng, H. Mai, J. Zhao, and X. Liu, "Cancer image classification based on DenseNet model," Journal of Physics: Conference Series, vol. 1651, no. 1, p. 012143, 2020.