Schaferct: Accurate Bandwidth Prediction for Real-Time Media Streaming with Offline Reinforcement Learning

Qingyue Tan[†§], Gerui Lv[†§], Xing Fang[¶§], Jiaxing Zhang[†§], Zejun Yang[†§], Yuan Jiang[†§], Qinghua Wu[†§]
[†] Institute of Computing Technology, Chinese Academy of Sciences
[¶] Institute of Automation, Chinese Academy of Sciences
[§] University of Chinese Academy of Sciences

Grand Challenge

Goal: Developing a deep learning-based policy model (receiver-side bandwidth estimator, π) with offline RL techniques to improve QoE for RTC system users as measured by objective audio/video quality scores.

Given: Dataset of trajectories for Microsoft Teams audio/video calls.

- Training dataset: 18859 calls.
- Evaluation dataset: 9405 calls containing ground truth (bottleneck link bandwidth).

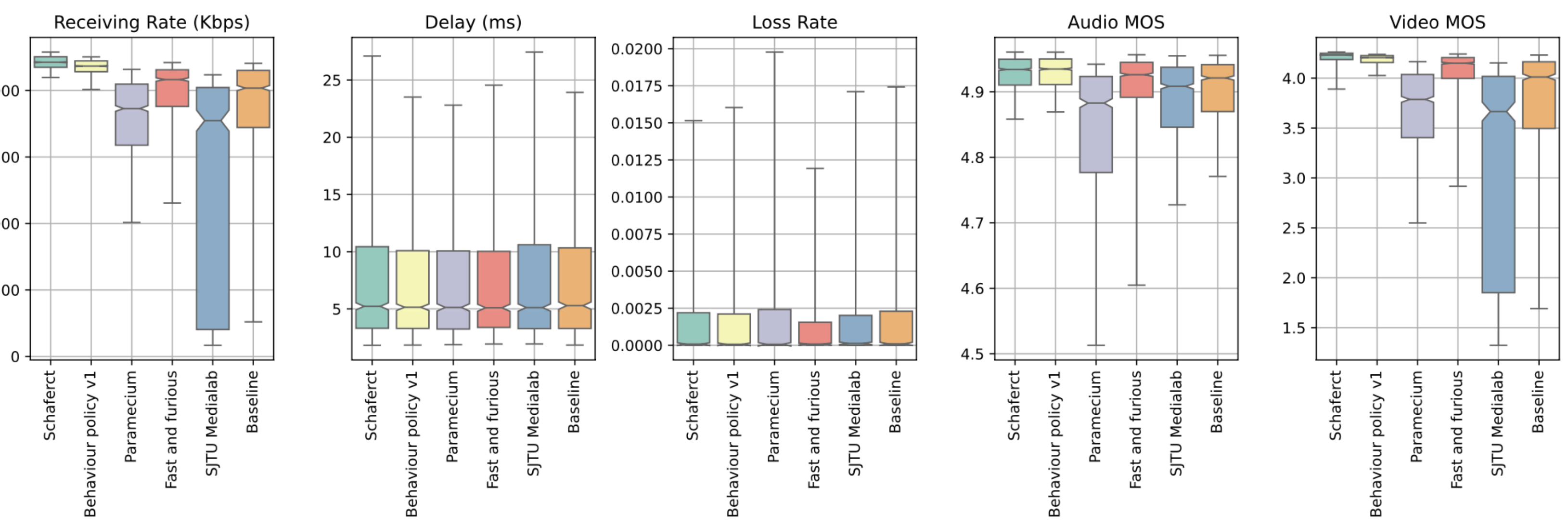
Evaluation: The scores in 2-stage evaluation. The scoring function:

$$\mathbb{E}_{call\ legs}\left[\mathbb{E}_n\left[r_n^{audio}+r_n^{video}\right]\right]\ \epsilon\ [0,10]$$

Team Schaferct: We won the 🏆 **first prize** in ACM MMSys 2024 Grand Challenge on Offline Reinforcement Learning for Bandwidth Estimation in Real Time Communications.

Results: Conducted by Grand Challenge Committee

Our model, **Schaferct**, demonstrates comparable performance to the best behavior policy (v1) in the released datasets across all metrics.

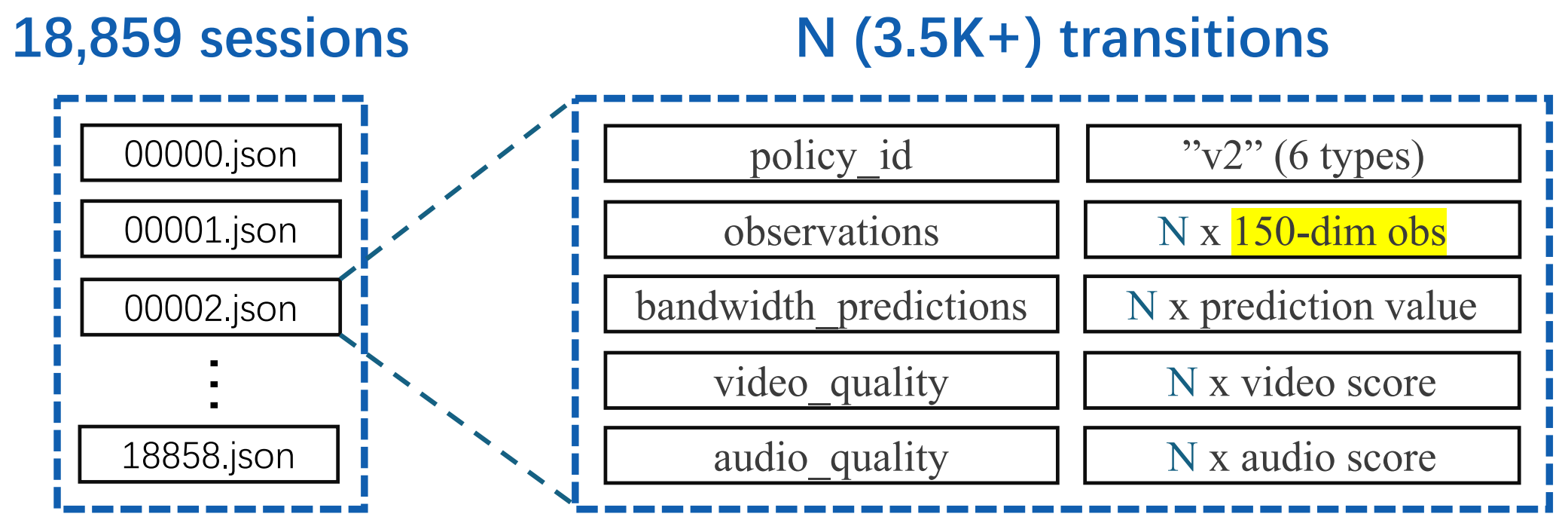


Results: 🌟 Final Evaluation Stage Ranking

In real-world test (600 3-minute calls) across diverse network conditions with temporal fluctuations, Schaferct took the first place with the highest scores.

Rank	Model	Score	95% CI
1	**Schaferct**	8.93	[8.88, 8.97]
2	**Fast and furious**	8.70	[8.65, 8.76]
3	Paramecium	8.34	[8.28, 8.39]
4	SJTU Medialab	7.89	[7.82, 7.96]

Dataset

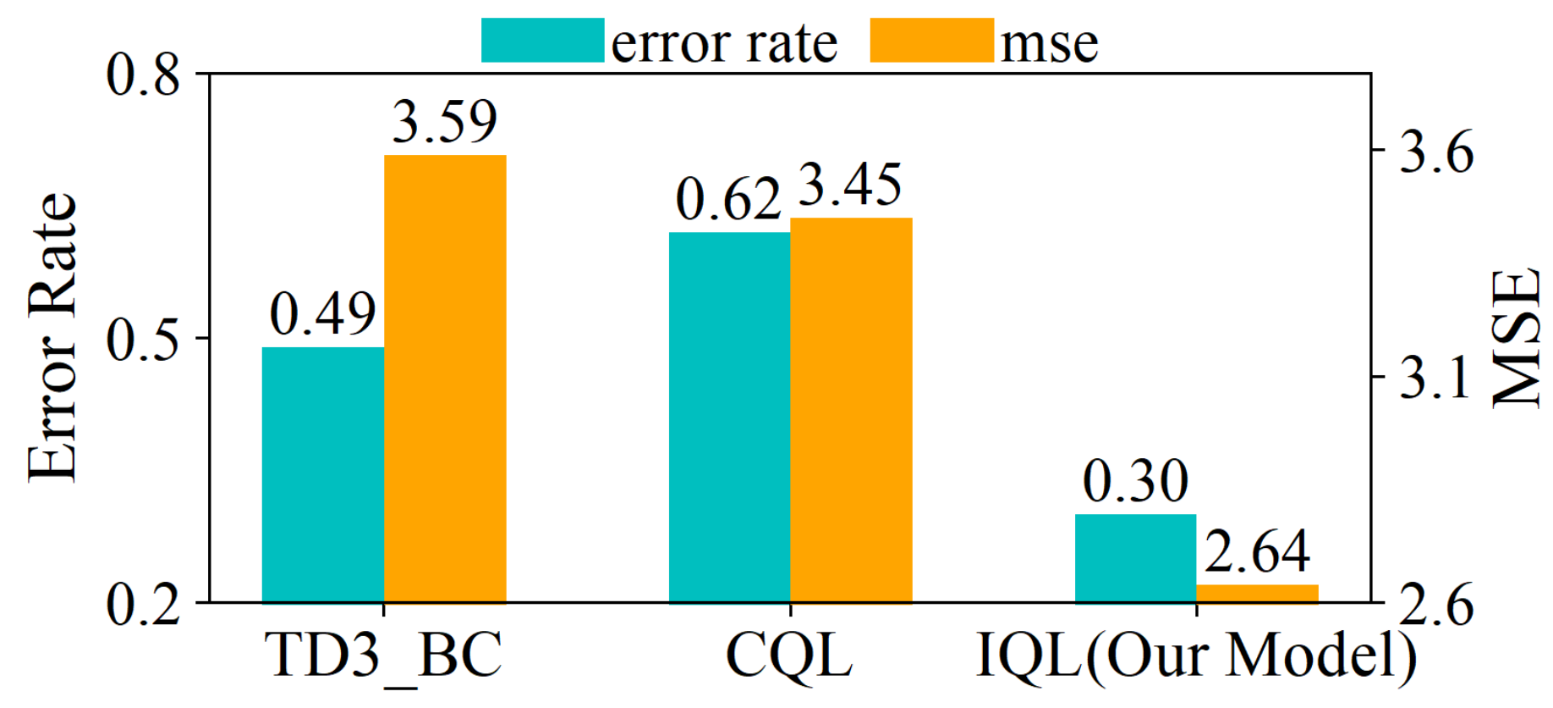


150-dim obs: 15 Features X [5 Long MI (600ms) + 5 Short MI (60ms)]

1	Receiving rate	6	Minimum seen delay	11	Packet loss ratio
2	Number of received packets	7	Delay ratio	12	Average number of lost packets
3	Received bytes	8	Delay average minimum difference	13	Video packets probability
4	Queuing delay	9	Packet interarrival time	14	Audio packets probability
5	Delay	10	Packet jitter	15	Probing packets probability

Design Choice 1: Offline RL Algorithm

The main challenge in offline RL is trading off policy improvement against distributional shift.



How other methods address this problem:

- TD3+BC: Constrains the policy to limit how far it deviates from the behavior policy.
- CQL: Regularizes the learned value functions to assign low values to out-of-distribution actions.

Implicit Q-Learning (IQL) solves this by never needing to directly query or estimate values for actions that were not seen in the data, it uses the expectile regression update method to approximate the optimal value function. As shown in our local evaluation results, IQL has the lowest MSE and error rate, so we eventually choose IQL to train our model.

Design Detail of IQL

In the policy evaluation stage, IQL approximate the optimal value function $V(s)$ with the asymmetric loss L_2^τ :

$$\mathcal{L}_V(\psi)=\mathbb{E}_{(s,a)\sim\mathcal{D}}[L_2^\tau(\,Q_{\hat{\theta}}(s,a)\ -\ V_\psi(s)\,)]$$

$$L_2^\tau(u)=|\tau\ -1(u<0)|u^2$$

The state-action value function $Q_\theta(s,a)$ is updated by minimizing the temporal difference (TD) loss:

$$\mathcal{L}_{\mathrm{Q}}(\theta)=\mathbb{E}_{(s,a,s')\sim\mathcal{D}}[(r(s,a)+\gamma V_\psi(s')-Q_\theta(s,a))^2]$$

In the policy extraction stage, IQL optimizes the final policy $\pi_\phi(s)$ by minimizing the following loss:

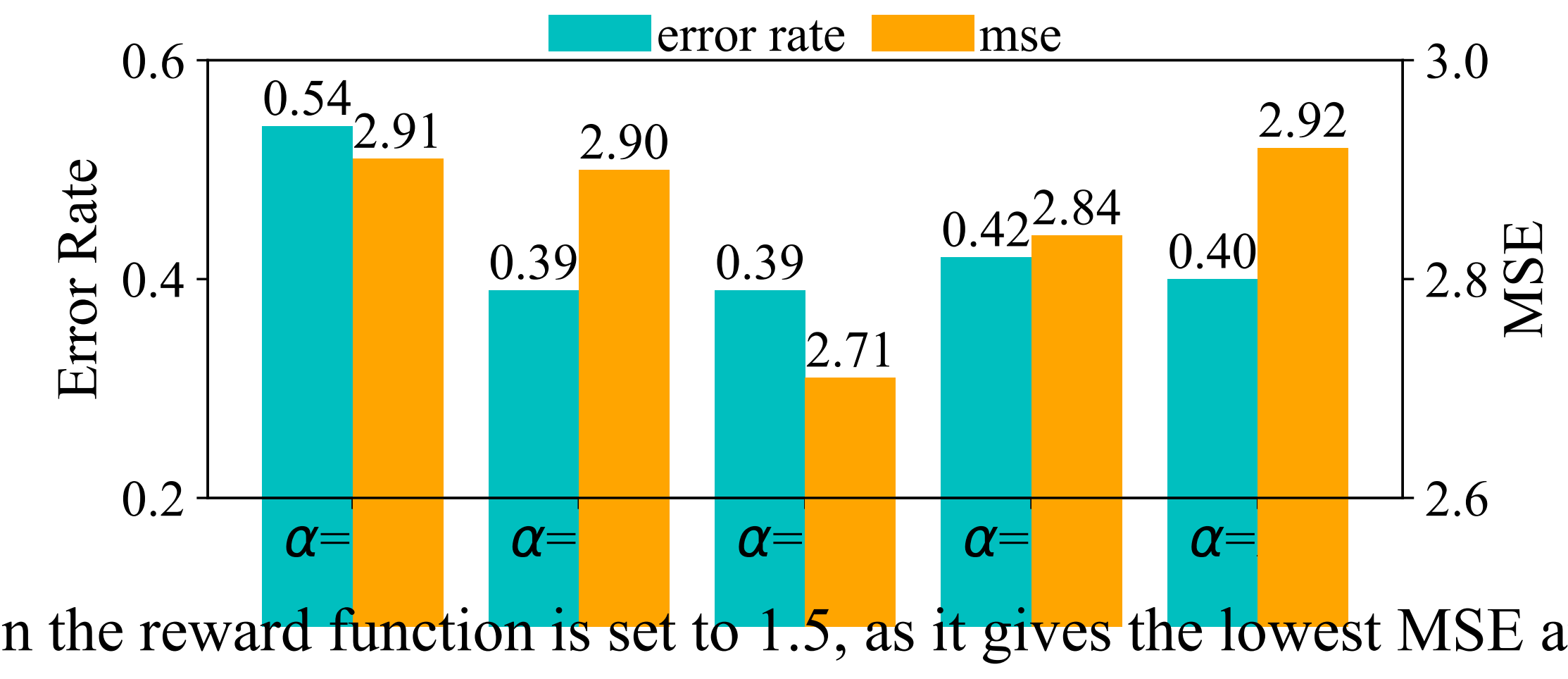
$$\mathcal{L}_\pi(\phi)=\mathbb{E}_{(s,a)\sim\mathcal{D}}[\exp(\,\beta\left(Q_{\hat{\theta}}(s,a)-V_\psi(s)\right)\ \log\pi_\phi\,(a|s)\,)]$$

Design Choice 2: State, Action and Reward

State: Normalized 150-dimensional observation (raw features).

Action: Predicted bandwidth in Mbps.

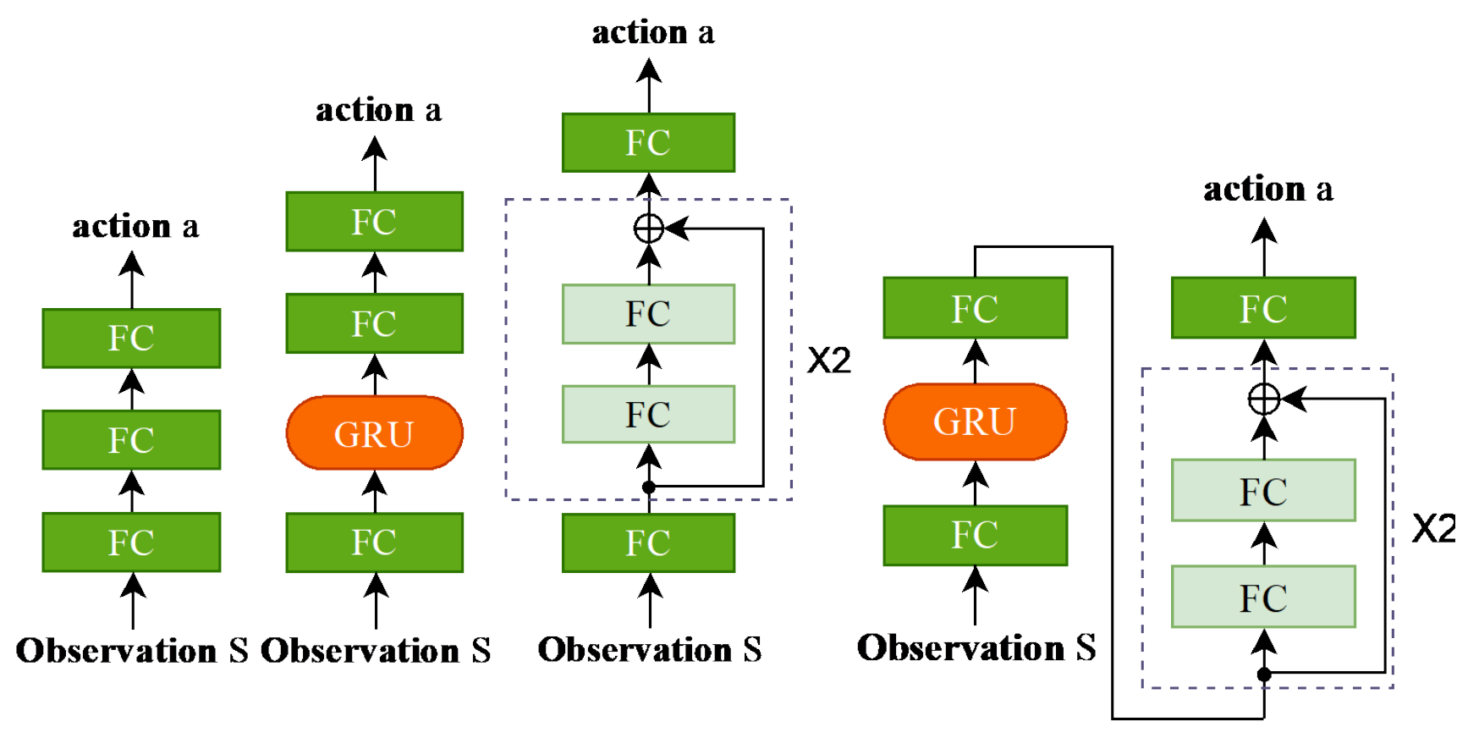
Reward: Both audio and video objective scores are used, with different weight, the reward function is defined as: $r(s,a)=(2-\alpha)*q_a+\ \alpha*q_v$

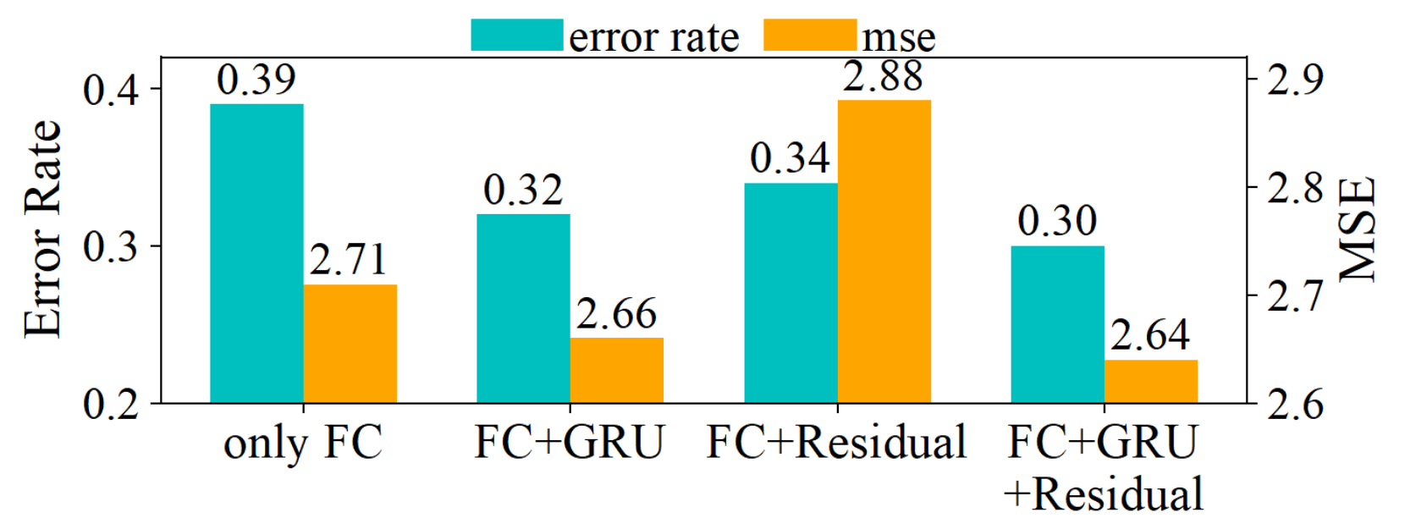


The weight α in the reward function is set to 1.5, as it gives the lowest MSE and error rate.

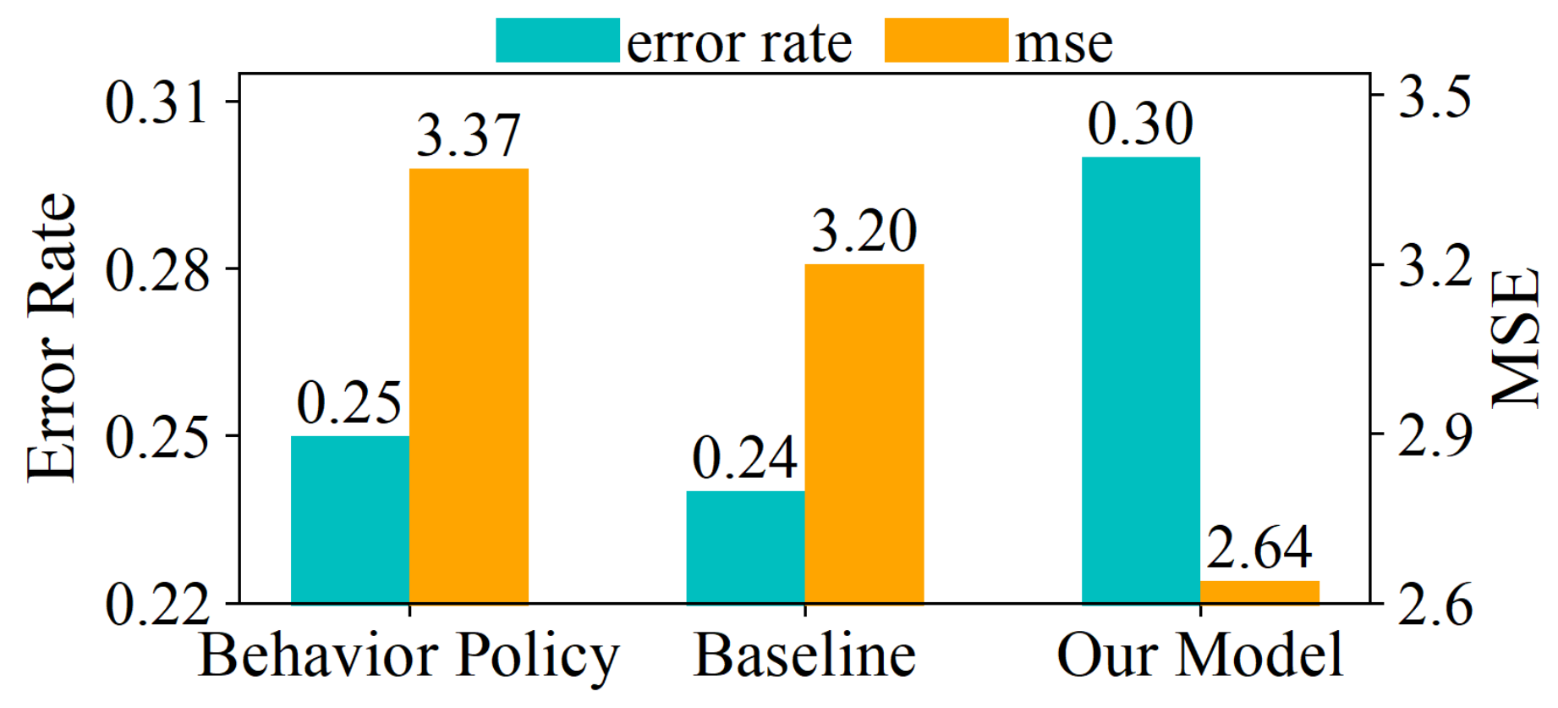
Design Choice 3: Actor Network Structure

We try out 4 different actor network structures: 1) only FC; 2) FC+GRU; 3) FC+Residual and 4) FC+GRU+Residual. We take the last one as our model for it has the lowest MSE.





Evaluation: Prediction Accuracy



Our model has the lowest MSE yet the highest error rate. We infer that MSE is more representative than error rate, as the latter is constrained in $[0,1]$ while the former is not.

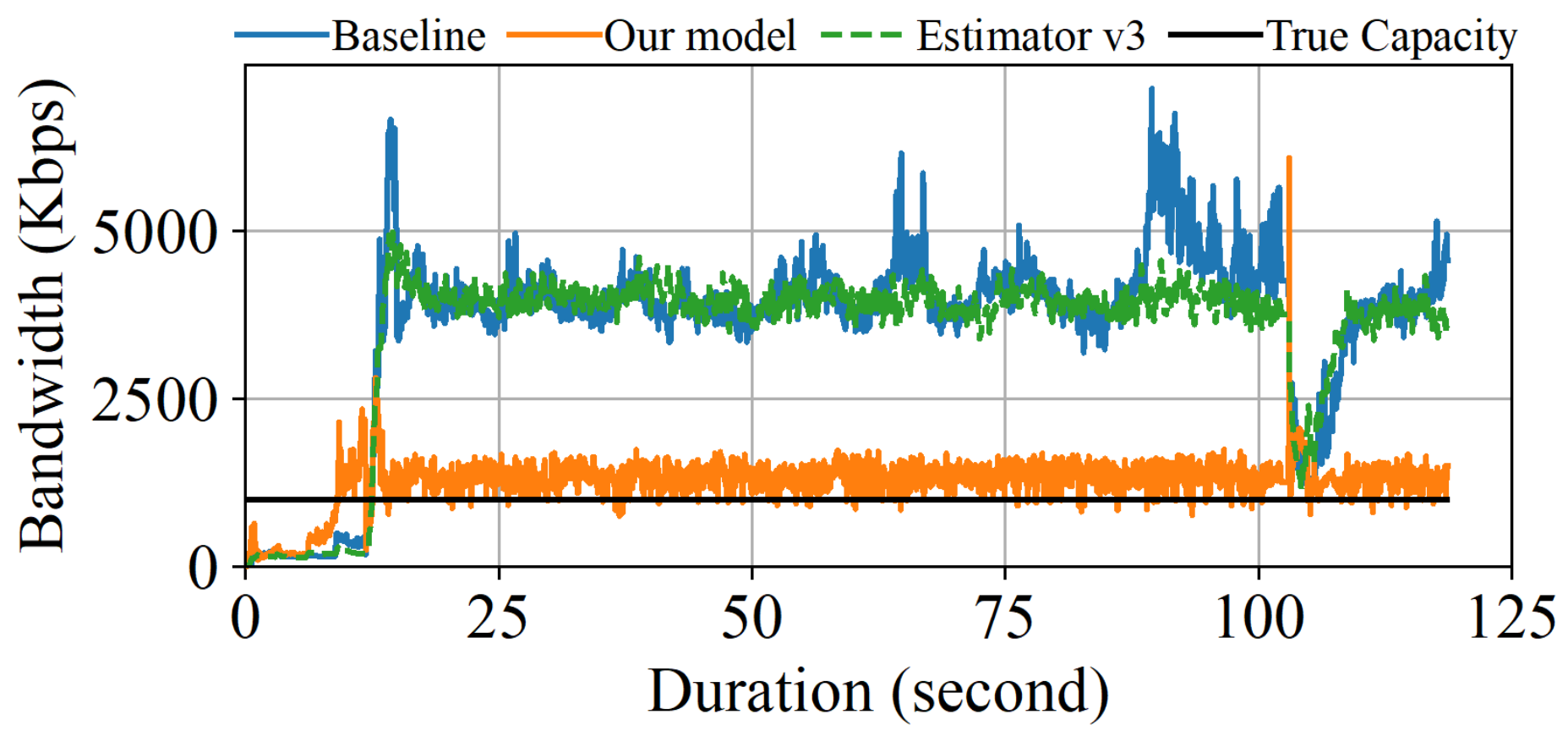
Evaluation: Case Study

Case #1:

The behavior policy: significantly overestimates the bottleneck link bandwidth.

The baseline model: follows the behavior policy, ends up in overestimation.

Our model: closely aligns with the true capacity.

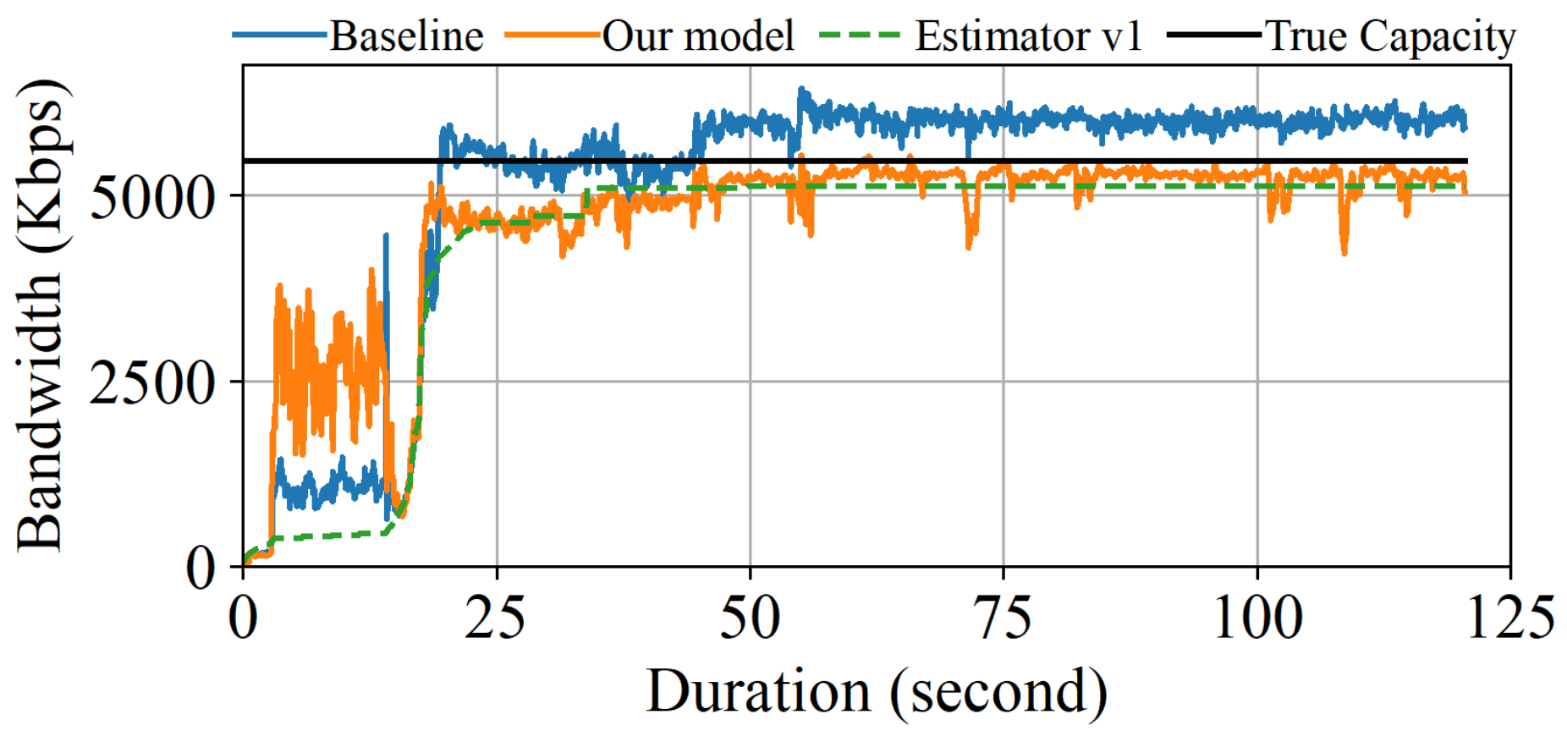


Case #2:

The behavior policy: predicts the bandwidth just fine.

The baseline model: overestimates the ground truth after start-up phase.

Our model: aligns with the behavior policy with more conservative and accurate predictions.



Limitations

- **Dataset**: Only 1,800 sessions are used for training due to the hardware constraints (e.g., GPU memory size) in our training environment.
- **Session Selection**: Session selection is random, without considering the distribution of observation-action-reward.
- **Feature Engineering**: All metrics are used.

Conclusion

- We evaluated different offline RL algorithms and finally chose IQL to train our model.
- We conducted multiple experimental studies, including redesigning the actor network architecture and selecting appropriate parameter values.
- Our model reduced MSE by 18%-22% compared to both the baseline and six behavior policies, and won the first prize of the Bandwidth Estimation Challenge at ACM MMSys 2024.