

Human-in-the-Loop Bandwidth Estimation for Quality of Experience Optimization in Real-Time Video Communication

Sami Khairy, Gabriel Mittag, Vishak Gopal, Ross Cutler

Microsoft

Abstract

The quality of experience (QoE) delivered by video conferencing systems is significantly influenced by accurately estimating the time-varying available bandwidth between the sender and receiver. Bandwidth estimation for real-time communications remains an open challenge due to rapidly evolving network architectures, increasingly complex protocol stacks, and the difficulty of defining QoE metrics that reliably improve user experience. In this work, we propose a deployed, human-in-the-loop, data-driven framework for bandwidth estimation to address these challenges. Our approach begins with training objective QoE reward models derived from subjective user evaluations to measure audio and video quality in real-time video conferencing systems. Subsequently, we collect roughly 1M network traces with objective QoE rewards from real-world Microsoft Teams calls to curate a bandwidth estimation training dataset. We then introduce a novel distributional offline reinforcement learning (RL) algorithm to train a neural-network-based bandwidth estimator aimed at improving QoE for users. Our real-world A/B test demonstrates that the proposed approach reduces the subjective poor call ratio by 11.41% compared to the baseline bandwidth estimator. Furthermore, the proposed offline RL algorithm is benchmarked on D4RL tasks to demonstrate its generalization beyond bandwidth estimation.

1 Introduction

By transforming how people connect, collaborate, and communicate across physical barriers and geographical divides, video conferencing systems have become vital for maintaining global business operations and providing accessible education (Markudova and Meo 2023; Eo et al. 2022). The quality of experience (QoE) offered by these systems, which is a measure of a user’s overall satisfaction with a multimedia conferencing system, is partly dependent on the estimation of the available bandwidth, which is defined as the bottleneck link’s unused capacity between a sender and receiver that varies over time due to fluctuations in concurrent traffic (Strauss, Katabi, and Kaashoek 2003). As illustrated in Figure 1, the receiver client estimates the available bandwidth from packet-level statistics and feeds this information back to the sender. In real-time communication (RTC) systems, this estimate guides the selection of target bitrates for the

Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

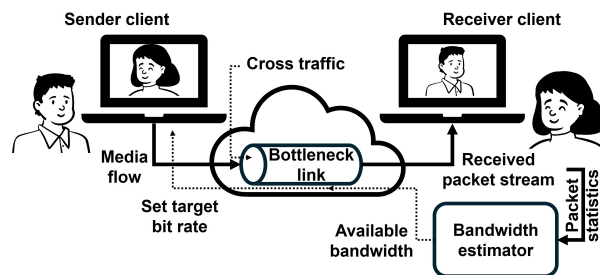


Figure 1: Bandwidth estimation in RTC: two endpoints exchange media over a time-varying bottleneck link. Cross-traffic reduces the available bandwidth. The receiver client infers available bandwidth from packet statistics (rate, delay, jitter, loss), which is fed back to the sender client to set encoder target bitrates. Accurate estimation is critical for optimizing QoE, as it directly impacts video smoothness, audio clarity, and overall user satisfaction.

audio and video encoders, thereby regulating the sender’s transmission rate (Li et al. 2022; Wang et al. 2021). Overestimating the available bandwidth results in network congestion, as the client transmits data at a rate higher than the network can handle (Zhang et al. 2020). Network congestion is characterized by increased delays in packet delivery, jitter, and potential packet losses, which manifest for remote meeting users as frequent resolution changes, video freezes, garbled speech, and audio/video desynchronization (Bentaleb et al. 2022; Zhang et al. 2020). Conversely, underestimating the available bandwidth causes the client to encode and transmit audio/video streams at a lower rate than the network can actually support, leading to under-utilization and sub-optimal QoE. Accurately estimating available bandwidth is therefore crucial for delivering optimal QoE to users in RTC systems.

In practice, however, estimating the available bandwidth presents numerous technical and design challenges. Firstly, because of routing decisions, traffic policing, and traffic shaping mechanisms implemented by internet service providers and cloud infrastructure, network paths between senders and receivers in video conferencing systems are highly dynamic and carry fluctuating traffic loads. Secondly, there are various first and last-mile networking technologies,

such as cellular (3G, 4G, 5G), Wi-Fi, WiMAX, and wired connections, each with distinct packet transmission characteristics, further complicating the estimation task. Thirdly, traffic from different applications sharing the same bottleneck link often uses different transport protocols with varying fairness mechanisms and competition dynamics, adding another layer of complexity. Together, these factors create a partially observable environment for a video conferencing client, which can only access local packet statistics to infer available bandwidth (Khairy et al. 2024). While all estimators must be designed with these constraints in mind, evaluating and improving them requires defining end-to-end (E2E) metrics that truly capture QoE, rather than relying solely on traditional Quality of Service (QoS) metrics such as throughput, delay, and packet loss, whose correlation with QoE is context-dependent and not well understood (Khairy et al. 2024). By focusing on E2E QoE metrics, we ensure system enhancements align with user-perceived quality, leading to more meaningful improvements. This motivates a data-driven approach that learns directly from user-aligned signals rather than proxying through QoS alone.

In this work, we propose a holistic data-driven framework for designing next-generation available bandwidth estimators suitable for real-world deployment. Specifically, our contributions are as follows.

1. First, we train objective QoE reward models, which measure E2E audio and video quality. These models predict mean audio and video quality scores based on subjective user evaluations following ITU-T P.808 and P.910 guidelines.
2. Second, we curate a comprehensive training dataset by collecting roughly 1M network traces annotated with QoE rewards from Microsoft Teams calls. In these calls, clients used a deployed unscented Kalman filter (UKF) for baseline bandwidth estimation.
3. Third, we develop a novel distributional offline reinforcement learning (RL) algorithm to train a neural network-based bandwidth estimator optimized for QoE. The proposed algorithm extends the state-of-the-art Implicit Q-learning (IQL) algorithm (Kostrikov, Nair, and Levine 2021) to the distributional RL paradigm to improve robustness, and employs asymmetric learning signals for the actor and critic based on domain knowledge.
4. Finally, we conduct extensive testbed and real-world evaluations, demonstrating significant improvements in objective QoE metrics and subjective ratings in large-scale A/B tests in production. Specifically, it is shown that the proposed approach reduces the subjective poor call ratio by 11.41% compared to the baseline estimator.
5. In addition, to assess generalization beyond the available bandwidth estimation domain, we benchmark the proposed offline RL algorithm on standard continuous control tasks from the D4RL benchmark suite (Fu et al. 2020), showing competitive performance with state-of-the-art methods.

This work is deployed in production within the Microsoft Teams real-time media stack, serving millions of daily active

users across diverse network conditions and device classes. By combining human-in-the-loop QoE modeling with offline RL, our approach closes the gap between offline policy optimization and safe large-scale deployment in latency-sensitive systems. Beyond RTC, the methodology can generalize to other networked multimedia applications where real-time resource allocation is critical. The next section reviews prior work on bandwidth estimation, QoE-driven optimization, and offline RL in networking.

2 Related Work

2.1 Bandwidth Estimation in RTC

In RTC systems, available bandwidth refers to the bottleneck link capacity between a sender and a receiver, minus traffic from competing flows. This quantity fluctuates dynamically due to cross-traffic variations, routing changes, and link-layer dynamics. Accurate bandwidth estimation is critical because it drives the audio/video encoder’s target bitrates. Overestimation leads to congestion and packet loss, while underestimation wastes capacity and reduces perceptual quality (Bentaleb et al. 2022; Zhang et al. 2020).

Traditional bandwidth estimation schemes such as GCC (Carlucci et al. 2016), NADA (Zhu et al. 2020), and SCReAM (Johansson et al. 2024) are built on fixed heuristics or model-based filters that react to network-level indicators like packet delay, loss, or throughput trends. While these methods are widely deployed, they are often tuned for conservative performance and can underutilize capacity in variable conditions. Their reliance on QoS metrics as optimization targets is a key limitation: QoS does not always align with user-perceived QoE (Engelke and Zepernick 2007). QoE is influenced by complex interactions between network behavior, codec adaptation, and human perception. As a result, estimators optimized for QoS may fail to maximize actual user satisfaction, particularly in heterogeneous environments with diverse access technologies. This misalignment motivates estimators that are trained and evaluated using QoE-aligned objectives rather than QoS proxies.

2.2 Online RL for Bandwidth Estimation

RL enables agents to learn control policies by interacting with an environment to maximize cumulative rewards. Widely used continuous-control algorithms include Proximal Policy Optimization (PPO) (Schulman et al. 2017), Deep Deterministic Policy Gradient (DDPG) (Lillicrap et al. 2015), and Soft Actor-Critic (SAC) (Haarnoja et al. 2018). These methods have been explored for adaptive rate control in RTC as a way to replace or augment rule-based controllers. For example, R3Net (Fang et al. 2019) trained an RL agent to adjust sending rates from packet statistics, and BoB (Bentaleb et al. 2022) integrated RL fine-tuning into an existing estimator.

However, online RL requires extensive exploration during training, which is unsafe in production because suboptimal actions can harm user experience. To avoid this, training is often conducted in network simulators or emulators, including our prior work (Gottipati et al. 2023) and other studies that remain confined to controlled environments (Fang

et al. 2019; Bentaleb et al. 2022). While such platforms enable repeatable experiments, they cannot capture the full diversity of real-world cross-traffic, devices, and access networks. This mismatch, often referred to as the simulation-to-reality gap, limits the transferability of policies trained purely in artificial environments. A notable exception is OnRL, which demonstrated end-to-end online policy learning with a QoS-based reward for mobile video telephony under tightly controlled safeguards and a constrained action space (Zhang et al. 2020). While this establishes feasibility in a focused domain, sustaining exploration at enterprise RTC scale across heterogeneous networks and devices remains risky. By contrast, our framework trains QoE-aligned reward models from subjective studies and uses them directly for offline policy optimization on large-scale real-world logs.

2.3 Offline RL for Bandwidth Estimation

Offline RL removes the need for live exploration by learning policies from pre-collected datasets. This makes it appealing for bandwidth estimation in RTC, where safety and predictability are paramount. Nevertheless, offline RL faces challenges such as distribution shift, in which learned policies select actions that are not observed in the dataset, and overestimation bias for out-of-distribution actions. Algorithms like Conservative Q-Learning (CQL) (Kostrikov, Nair, and Levine 2021), IQL (Fujimoto and Gu 2021a), and advantage-weighted approaches (Peng et al. 2019; Ashvin et al. 2020; Kozakowski et al. 2022) address these issues.

For bandwidth estimation, Mowgli (Agarwal et al. 2025) trained rate-control policies from passive telemetry, outperforming GCC without online training. The ACM MM-Sys’24 offline RL for bandwidth estimation grand challenge (Khairy et al. 2024) released a dataset collected in a controlled testbed and emulation environment with QoE-derived reward labels, enabling training of human-aligned offline bandwidth estimation policies but not yet reflecting large-scale production diversity (Lu et al. 2024; Zhang, Tao, and Wang 2024; Cetinkaya et al. 2024; Gottipati et al. 2024). The present paper advances this trajectory by pairing QoE-aligned rewards with a distributional RL agent trained on large-scale Teams telemetry and by validating the learned policy in production A/B tests.

2.4 Motivation and Gaps

Building on the limitations of rule-based controllers in heterogeneous networks, the risks of online training, and the promise of offline learning, we distill four deployment-driven challenges:

- **Dynamic and heterogeneous networks:** network conditions vary widely across users and over time, making it difficult for static heuristics to perform well universally.
- **Partial observability:** the true state of the network is not directly measurable; estimators must infer it from noisy and delayed observations.
- **QoE alignment:** traditional metrics like throughput and packet loss do not always correlate with user satisfaction.

Estimators must optimize for perceptual quality metrics which correlates with mean opinion scores (MOS).

- **Safety and deployability:** online RL poses risks to user experience during training. Offline RL mitigates this by learning from historical data, enabling safe deployment.

Our design addresses these challenges in an integrated way. To cover network diversity, we train on large-scale Microsoft Teams telemetry. To mitigate partial observability, we construct compact, history-aware state from local packet timing, loss, and jitter that captures network dynamics. To align optimization with user experience, we train audio and video QoE models on subjective evaluations conducted under ITU-T P.808/P.910 and use their predictions as rewards. To preserve safety in deployment, we learn policies offline and A/B test before broad exposure.

We implement this blueprint in a deployed system that couples a human-in-the-loop data pipeline with a distributional RL agent trained offline and executed within the media stack at millisecond-scale latency. The next section details the system and learning algorithm, followed by testbed and production evaluations.

3 A Data-Driven Framework for Human-Aligned Bandwidth Estimation

3.1 QoE Reward Modeling from Human Feedback

The use of proper reward functions that align with a user’s experience of audio and video quality is crucial when training and evaluating bandwidth estimation models. Reward functions that accurately reflect user experience ensure that the models prioritize the aspects of quality that matter most to users, such as clarity, smoothness, and minimal latency. This alignment is essential because it directly impacts user satisfaction and engagement. For instance, a model that optimizes for technical metrics without considering user experience may result in high bandwidth usage without a corresponding improvement in perceived quality. Therefore, incorporating user-centric reward functions helps in developing models that not only perform well in technical evaluations but also enhance the overall user experience.

Audio quality model A signal-based audio quality model was initially trained to predict the quality of received audio signals in peer-to-peer (P2P) calls. Specifically, P2P calls were conducted between pairs of machines connected through networking emulation software that emulated various network characteristics, such as burst loss, traffic policing, and bandwidth fluctuations. Due to these network transmission characteristics, the received audio signals were often distorted and corrupted with random noise, the nature of which depended on the emulated network scenario. The dataset of received audio recordings was rated using subjective scores according to ITU-T Recommendation P.808. This process involved human raters listening to the audio samples and assigning values on a scale from 1 to 5, with 5 representing the best quality. Subsequently, a no-reference Wav2Vec-based model was trained on this dataset, and achieved a high Pearson Correlation Coefficient (PCC) of 0.951 and a Root

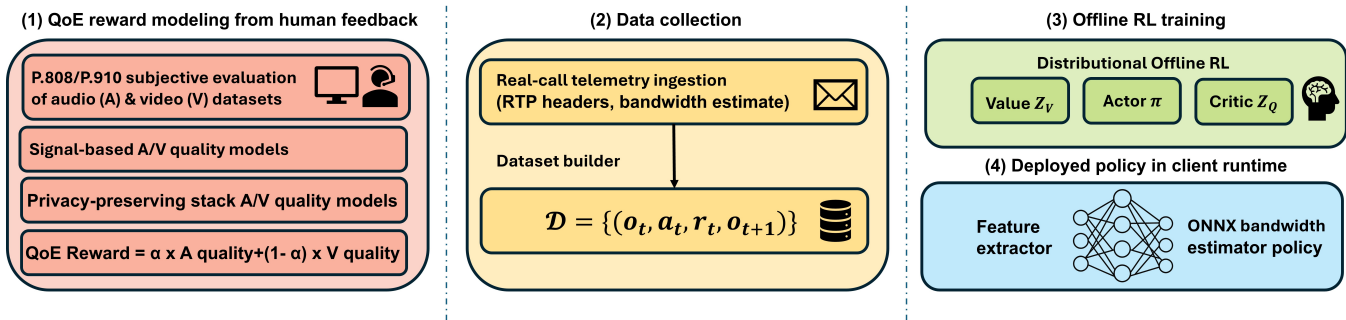


Figure 2: Overall framework: (1) QoE reward modeling from human feedback (P.808/P.910) produces in-stack audio/video quality predictors for use as a QoE reward; (2) real-call telemetry (RTP headers, baseline bandwidth estimates) is transformed into (o_t, a_t, r_t, o_{t+1}) trajectories; (3) a distributional offline RL agent (value Z_V , critic Z_Q , policy π) is trained and exported to ONNX for client-side inference; (4) telemetry and A/B testing close the deployment loop.

Mean Square Error (RMSE) of 0.194 with the subjective P.808 scores in the validation set.

While this signal-based model proved effective in laboratory experiments due to its ability to leverage detailed audio signal features, it was not practical for deployment in real-world real-time client environments due to its high complexity and the need for raw audio signals which violates user privacy. To address these issues, the signal-based model was distilled into a stack audio quality model that could run efficiently and cost-effectively in the media stack. The stack audio quality model utilizes key media metrics such as audio receive rate, jitter, and packet loss concealment, achieving a high PCC of 0.972 with objective audio quality scores. This transition enabled real-world, real-time, privacy-preserving audio-quality evaluation without compromising accuracy.

Video quality model In a similar setup, an LSTM-based full-reference video quality model that leverages VMAF (Li et al. 2016) and freeze features to capture temporal distortions is developed (Mittag et al. 2023). While this model demonstrated a high PCC of 0.985 and an RMSE of 0.202 with subjective data provided by human raters according to ITU-T Recommendation P.910, it was also not suitable for client-side deployment due to its high complexity and the need for full video reference data. To overcome these limitations, the full-reference video quality model was distilled into a stack video quality model that relies on media metrics such as resolution, quantization parameters, the user’s viewport size, freezes, and frame rate (FPS). This stack model achieved an extremely high PCC of 0.998 with objective video quality scores, demonstrating its effectiveness in real-time applications while maintaining low computational overhead. The use of media metrics instead of raw video data ensures that the model can operate efficiently on client devices, preserving user privacy and reducing resource consumption.

QoE reward model Given the stack audio and video quality modes, we define the QoE reward as

$$r_{\text{QoE}} = \alpha \times \text{audio quality} + (1 - \alpha) \times \text{video quality}, \quad (1)$$

where $\alpha \in [0, 1]$ is a weighting parameter. This formulation allows for a balanced consideration of both audio and video quality, reflecting the overall user experience during multimedia interactions. By adjusting the parameter α , the model can prioritize either audio or video quality based on specific application requirements or user preferences. Such a QoE reward model aligns closely with users’ subjective experiences, as it integrates key perceptual metrics from both audio and video streams. Optimizing this reward model can lead to significant improvements in user satisfaction, as it ensures that both audio and video quality are maintained at optimal levels, thereby enhancing the overall multimedia experience.

3.2 Data Collection

Training bandwidth estimation models on real-world logs offers a robust alternative to lab-emulated datasets. This approach captures detailed network traces from actual calls, enabling models to learn from realistic conditions which would be otherwise hard to emulate. These conditions include dynamic network conditions, network anomalies, user behaviors (e.g., users joining or leaving a call, switching cameras on or off, and talking or listening), and different machine hardware/software specification. In our production system, a rule-based estimator based on an Unscented Kalman Filter (UKF) is deployed. Similar to WebRTC (Bergkvist et al. 2012), UKF models network delay dynamics and adapts bandwidth estimates using static functions derived from state variables such as one-way delays, delay gradients, and loss rates. For example, it scales estimates in response to changes in delay. While extensively validated in production, the UKF’s reliance on pre-defined heuristics limits its adaptability to evolving network conditions, making it a strong baseline but not a complete solution. The collected logs include RTP packet headers (Schulzrinne et al. 2003), UKF bandwidth estimates, and predicted audio/video quality scores by the in-stack models. These logs are transformed into trajectories containing observations, actions, and rewards suitable for offline RL. In total, we have collected approximately 1M traces, which yield about 1.25 trillion state-action-reward pairs in the RL

setting.

3.3 Bandwidth Estimation with Offline RL

Estimating available bandwidth in real-time video conferencing is inherently challenging due to the unobservable nature of the bottleneck link between a sender and a receiver. The agent must infer bandwidth from noisy and stochastic signals derived from the received packet stream, which are influenced by cross-traffic, queueing dynamics, and other network uncertainties. We formulate bandwidth estimation as a Partially Observable Markov Decision Process (POMDP) and propose a novel training algorithm tailored to real-time video conferencing scenarios.

Bandwidth estimation as a POMDP

- **State space:** the underlying network state includes link capacity, cross-traffic load, propagation delay, packet loss, and jitter. These factors are influenced by external conditions such as mobility, signal interference, and transport technology (e.g., 5G, satellite), making the true state unobservable and dynamic.
- **Observation space:** observations are computed from received RTP packet headers over both short-term (60ms) and long-term (600ms) monitoring intervals (MIs). At each time step, the observation vector aggregates nine key network features: receiving rate, one-way delay, packet loss ratio, packet jitter, probabilities of video, audio, screen share, and probing packets, as well as the latest probing bandwidth estimate across 3 short-term and 3 long-term MIs. This design captures both immediate and longer-term network behaviors, providing a comprehensive, multi-scale view of network performance for robust bandwidth estimation.
- **Action space:** the agent’s action is the available bandwidth estimate in bits-per-second (bps), which is used to set the target bit rate for media encoders at each decision step.
- **Reward function:** rewards reflects the predicted QoE for a given state-action pair, as defined in Eq. (1).

Asymmetric actor-critic To address the challenges of partial observability and temporal dynamics, we adopt an asymmetric actor-critic architecture. The actor network uses an LSTM module to capture temporal patterns from recent observations, enabling adaptive bandwidth predictions. The critic network, implemented as an MLP, leverages stacked historical features to estimate long-term QoE returns. This design balances responsiveness and stability: the actor focuses on immediate adaptation using recent data, while the critic benefits from broader temporal context. Empirically, we found this separation to improve training stability and policy performance in online evaluation.

Multi-modal actor-critic Bandwidth estimation presents a multi-modal learning challenge. The same observation may correspond to different bandwidth conditions depending on sender behavior, device capabilities, or media type. Similarly, QoE outcomes can vary across different devices

and user’s viewpoints even under identical network conditions. To model this complexity, we represent both the actor and critic as a mixture density network (MDN) parameterising a Gaussian mixture (GM). Specifically, each network has an output layer with $N \times 3$ neurons, where N is the number of components in the mixture. Each component $i \in [N]$ in mixture model is parameterized by the component weight in the mixture w_i , the component mean μ_i , and the component standard deviation σ_i . In this work, we set $N = 3$ unless otherwise mentioned.

Optimizing QoE with distributional offline RL To effectively optimize QoE in our bandwidth estimation setting, we develop a distributional offline RL algorithm based on IQL (Kostrikov, Nair, and Levine 2021). The proposed approach extends the standard IQL framework into the distributional RL paradigm (Bellemare, Dabney, and Munos 2017) by modeling the entire return distribution instead of just its expected value (Dabney et al. 2018b,a). By doing so, we capture the inherent uncertainty and multi-modal nature of network dynamics, leading to a more robust bandwidth estimation policy. Our algorithm is an asymmetric actor-critic method tailored for offline learning of bandwidth estimators: the critic learns the QoE return distributions (distributional Q and value functions), and the actor is optimized via advantage-weighted regression using different loss functions. We describe the key components of the proposed distributional IQL (DIQL) algorithm and training objectives below.

Distributional value Function ($V_\psi : s \rightarrow Z_V(s)$): rather than directly taking a max over actions, the value function $V_\psi(s)$ in IQL is learned to represent an upper envelope of the Q-function at state s using expectile regression. Since we now operate over return distributions rather than scalar values, we adopt a distributional formulation inspired by (Bellemare, Dabney, and Munos 2017). Specifically, the value network outputs a distribution over returns, parameterised as a GMM:

$$Z_V(s) \sim \sum_{i=1}^N w_i^V(s) \mathcal{N}(\mu_i^V(s), \sigma_i^V(s)),$$

where $w_i^V(s)$ are mixture weights and $(\mu_i^V(s), \sigma_i^V(s))$ denote the mean and standard deviation of the i -th Gaussian component. Similarly, the Q-value distribution for (s, a) is modelled as:

$$Z_Q(s, a) \sim \sum_{j=1}^N w_j^Q(s, a) \mathcal{N}(\mu_j^Q(s, a), \sigma_j^Q(s, a)).$$

To match these distributions, we require a metric suitable for GMMs. Following (Delon and Desolneux 2020), we consider the Mixture Wasserstein-2 (MW_2) distance:

$$MW_2^2(Z_V, Z_Q) = \min_{\lambda \in \Pi(w^V, w^Q)} \sum_{i=1}^N \sum_{j=1}^N \lambda_{ij} W_2^2(i, j),$$

where $\Pi(w^V, w^Q)$ is the set of couplings between mixture weights and $W_2^2(i, j)$ is the squared 2-Wasserstein distance

between Gaussian components:

$$W_2^2(i, j) = (\mu_i^V(s) - \mu_j^Q(s, a))^2 + (\sigma_i^V(s) - \sigma_j^Q(s, a))^2. \quad (2)$$

Computing the optimal coupling λ exactly is expensive. While the Sinkhorn algorithm with entropic regularisation (Cuturi 2013) can approximate it efficiently, it still requires many iterations. Instead, we adopt an upper bound using the independent coupling:

$$\widehat{MW}_2^2(Z_V, Z_Q) = \sum_{i=1}^N \sum_{j=1}^N w_i^V w_j^Q W_2^2(i, j), \quad (3)$$

which assumes independence between components. We find that this bound to work well in practice and avoids the computational overhead of iterative optimal transport solvers. Finally, to incorporate asymmetry, we modify the cost matrix by introducing an expectile-based weight $O(\tau, i, j)$ for each pair (i, j) using component means as support points, yielding the asymmetric loss:

$$L_V = \mathbb{E}_{(s,a) \sim \mathcal{D}} \left[\sum_{i=1}^N \sum_{j=1}^N w_i^V w_j^Q W_2^2(i, j) O(\tau, i, j) \right], \quad (4)$$

where

$$O(\tau, i, j) = |\tau - \mathbf{1}_{\{z < 0\}}|, \quad z = \mu_j^Q(s, a) - \mu_i^V(s).$$

In the single-component average return case where $(\sigma_i^V, \sigma_j^Q) \rightarrow 0$, this loss reduces to the standard IQL expectile loss (Kostrikov, Nair, and Levine 2021). In the general loss however, uncertainty in the value distribution can be captured through component variances. Minimising this objective pushes $Z_V(s)$ towards the τ -expectile of $Z_Q(s, a)$ while incorporating both location and scale information, providing a conservative estimate of the optimal value distribution without extrapolating to unseen actions.

Distributional critic ($Q_\theta : (s, a) \rightarrow Z_Q(s, a)$): The critic network predicts the distribution of cumulative discounted QoE returns for each state–action pair. We adopt a one-step distributional Bellman target:

$$Z_{\text{target}}(s, a) \doteq r(s, a) + \gamma Z_V(s'),$$

where $r(s, a)$ is the QoE reward (Eq. 1) and addition denotes a shift of the distribution by $r(s, a)$. Since $Z_V(s')$ is a Gaussian mixture $\{w_i, \mu_i, \sigma_i\}_{i=1}^N$, the target is also a mixture: $\{w_i, r + \gamma\mu_i, \gamma\sigma_i\}_{i=1}^N$. The critic Q_θ is trained to minimise the distributional TD error between its predicted distribution and $Z_{\text{target}}(s, a)$ using a symmetric component-wise squared 2-Wasserstein loss:

$$L_Q = \mathbb{E}_{(s,a,r,s') \sim \mathcal{D}} \left[\sum_{i=1}^N \sum_{j=1}^N w_i^{\text{target}} w_j^Q \Omega_2^2(i, j) \right], \quad (5)$$

where $\Omega_2^2(i, j) = (\mu_i^{\text{target}}(s, a) - \mu_j^Q)^2 + (\sigma_i^{\text{target}}(s, a) - \sigma_j^Q(s, a))^2$. This mirrors the (MW_2) -based approach in the value function but without the asymmetry, as the critic aims to match the Bellman target rather than form an upper envelope. Because V_ψ represents an upper expectile of Q ,

this loss biases $Z_Q(s, a)$ towards high-return actions in the dataset, enabling policy improvement. By modelling full distributions, the critic captures multi-modality in network scenarios where similar actions can yield both high and low QoE, avoiding the pitfalls of a single expected value. In practice, we employ two critics and select the mixture with the smaller mean to mitigate overestimation.

Actor policy function ($\pi_\phi : s \rightarrow Z_\pi(s)$): the policy network which serves as the bandwidth estimator, is also parametrized as a Gaussian mixture over the continuous action space $\pi_\phi(a|s)$. The actor is trained via advantage weighted regression (AWR) using the mean of the learned critic and value functions. Concretely, given a state s and an action a , the advantage function is defined as $A(s, a) = \bar{Q}_\theta(s, a) - \bar{V}_\psi(s)$, where $\bar{Q}_\theta(s, a) = \sum_{i=1}^N w_i^Q \mu_i^Q(s, a)$ and $\bar{V}_\psi(s) = \sum_{i=1}^N w_i^V \mu_i^V(s)$. Here \bar{Q}_θ and \bar{V}_ψ are mixture-mean expectations, consistent with the distributional parameterisation; we stop gradients through Q and V when updating π . The policy loss is,

$$L_\pi = -\mathbb{E}_{(s,a) \sim d} \left[\exp(\beta A(s, a)) \log \pi_\phi(a|s) \right], \quad (6)$$

where $\beta > 0$ is a temperature parameter. Equation (6) can be interpreted as a form of soft policy improvement: it trains π_ϕ to imitate actions in the dataset, but upweights those actions that would lead to higher-than-average returns according to the current Q estimate.

Implementation details The actor network consists of an LSTM layer with 128 neurons, followed by 5×128 dense layers with *tanh* activations. Dropout layers with a drop rate of 0.05 are introduced between dense layers for regularization. Output layer parameterises a 3-component GM: weights via softmax, means via *tanh* (bounded to $[-1, 1]$ but then scaled to bps as in Eq. 1 (Khairy et al. 2024)), and standard deviations via *softplus* with a small floor for stability. The architectures for the critic and value networks are identical except for the lack of the LSTM layer.

We train the agent on the collected dataset of real call traces, which have been pre-processed into sequences of (o_t, a_t, r_t, o_{t+1}) . Training proceeds by alternating the three updates described in the previous subsection. For each mini-batch sampled from the dataset: (i) update V_ψ by minimizing L_V (Eq.(4)), (ii) update π_ϕ by minimizing L_π (Eq.(6)), and (iii) update Q_θ by minimizing L_Q (Eq.(5)) with targets from the current V_ψ .

4 Evaluation

We evaluate our approach across three settings: (i) large-scale A/B testing in Microsoft Teams, (ii) controlled testbed experiments with diverse network profiles and ablations, and (iii) out-of-domain continuous control tasks to validate the proposed distributional offline RL algorithm.

4.1 Production A/B Testing

Training and deployment. We train a bandwidth estimation model based on the proposed framework using an Azure

NDm A100 v4 cluster for approximately a week. The resulting model is converted to ONNX and integrated into the Microsoft Teams media stack with the same feature normalization and preprocessing used in offline training to ensure parity. The estimator is invoked every 60ms with a median inference time of roughly 600 μ s which easily fits within real-time latency budgets.

Experimental design. A staged rollout was conducted to ensure safety: we first ran the estimator in shadow mode on a small cohort, then incrementally ramped to larger user populations. The final A/B test lasted two weeks, randomizing at the call level and treating over 25 million calls across diverse devices and network types at a global scale. For every call leg, the objective video and audio quality scores are reported from the deployed media stack models, as well as subjective poor call rate (PCR) if users submit a rating. PCR is computed over rated calls only. We report leg-wise means and relative deltas with 95% confidence intervals. All reported differences are significant at $p < 0.05$.

Metric	Treatment	Control	Delta	Delta%	P-value
Audio Quality Score	4.4702	4.4701	0.0001	+0.00%	0.0061
Video Quality Score	4.1721	4.1700	0.0020	+0.05%	7e-38
PCR	0.0160	0.0180	-0.0021	-11.41%	0.0224

Table 1: Production A/B outcomes. Audio/Video Quality Scores (\uparrow higher is better); PCR (\downarrow lower is better).

Results. Table 1 summarizes the outcomes. The proposed estimator consistently improves objective audio and video quality scores and reduces PCR by 11.41% over the baseline estimator. PCR was pre-registered as the primary user-facing Key Performance Indicator (KPI); the 11.41% relative reduction is statistically significant ($p < 0.05$) and reflects materially fewer subpar calls. Video quality score shows a small but statistically significant improvement of +0.05%, while the audio quality difference is statistically detectable yet practically negligible.

4.2 Testbed Experiments and Ablations

To complement the production study, we evaluate the trained bandwidth estimation model in a controlled emulation platform. Each evaluation run is a peer-to-peer video and audio call between two lab endpoints connected through a software network emulator that replays time-varying traces of capacity, delay, and loss. Profiles are deterministic and replayable. Each model is evaluated in 15 peer-to-peer video calls (two legs per call) and over 30 network profile, including burst loss (BL), random loss (RNDL), fixed bandwidth (FB), fluctuating bandwidth (FLB), and 4G scenarios. We report the average objective QoE reward and standard deviation per trace across all call legs.

Baselines and variants. We compare the proposed distributional offline RL algorithm to the following:

- **Behavior policy:** deployed rule-based bandwidth estimator used to collect logs; no learning.

- **Behavior cloning (BC):** a neural policy trained to imitate the UKF actions with a negative log-likelihood loss.
- **Implicit Q-Learning (IQL) (Kostrikov, Nair, and Levine 2021):** expectile-regressed value function; actor learned via advantage-weighted regression (AWR-style).
- **TD3BC (Fujimoto and Gu 2021b).** Actor maximizes $\mathbb{E}_{a \sim \pi}[Q(s, a)]$ with a behavior-cloning regularizer $\alpha \mathbb{E}_{(s, a) \sim \mathcal{D}}[\log \pi(a|s)]$. We sweep $\alpha \in \{1.0, 0.1, 0.01\}$ and report the best.
- **Quantile-regression critic (QR) (Dabney et al. 2018b):** a deep quantile network with 9 quantiles $\{0.1, 0.2, \dots, 0.9\}$ is first trained with quantile regression to model return distribution; actor extracted via advantage-weighted regression as in IQL. We also sweep the number of quantiles $\{3, 6, 9\}$ and report the best.

For all learned baselines and our method, we hold constant the observation space, policy/value network architectures, and hyperparameters. Only the method-specific parameters (e.g., IQL expectile, α for TD3BC, number of quantiles for QR) differ.

Training and selection. Each model is trained for 300 epochs (one full pass over the training set per epoch) with ADAM optimizer (Kingma and Ba 2015) with a learning rate of 3×10^{-5} and a batch size of 256 trajectories. Every 5 epochs, we compute the mean squared error (MSE) with respect to the behavior policy’s actions. The three checkpoints with the lowest MSE are evaluated online in the testbed and we report the best. This two-stage selection reduces variance and avoids over-fitting to offline metrics.

Results. In Table 2, methods within the top 1% of the best score in a network are typeset in **bold**. As Table 2 indicates, the proposed DIQL algorithm for training bandwidth estimation policies achieves the highest QoE across the majority of network profiles, being in the top 1% for 27/30 of profiles, with the largest improvements in lossy network conditions. Among all methods, DIQL achieves the highest average gain (0.0848), and the smallest minimum gain (-0.008), indicating its consistent improvement and minimal performance drop compared to the behavior baseline estimator. This means that improvements do not come at the cost of occasional severe regressions; guaranteeing safe deployment in real-world systems where even rare failures can significantly impact user experience.

4.3 D4RL Benchmark

Finally, we evaluate the distributional offline RL algorithm itself on standard MuJoCo tasks from the D4RL benchmark (Fu et al. 2020). We use three seeds per task, evaluate every 10,000 gradient steps, and train for 1M gradient steps with hyperparameters matching IQL. Architectures, batch size, optimizer, discount, target update rate, and state normalization are identical to IQL for parity, and actions are squashed to $[-1, 1]$ with the same policy parameterization. For the distributional value/critic networks, we use 3 components. Evaluation uses deterministic policies (mean action) over 100 episodes per checkpoint, and we report mean \pm std of

Network profile	Behavior	BC	IQL	TD3BC	QR	DIQL (ours)
FLB 1	2.377 ± 0.074	2.403 ± 0.022	2.405 ± 0.031	2.412 ± 0.024	2.423 ± 0.023	2.416 ± 0.029
FLB 2	2.510 ± 0.080	2.538 ± 0.075	2.659 ± 0.066	2.598 ± 0.060	2.581 ± 0.060	2.661 ± 0.060
FLB 3	2.068 ± 0.075	2.024 ± 0.117	2.028 ± 0.104	2.067 ± 0.078	2.013 ± 0.094	2.076 ± 0.103
FLB 4	2.418 ± 0.083	2.455 ± 0.066	2.480 ± 0.056	2.463 ± 0.076	2.455 ± 0.070	2.424 ± 0.064
BL 8ML25	3.441 ± 0.407	3.872 ± 0.134	3.865 ± 0.184	3.946 ± 0.142	3.858 ± 0.176	3.829 ± 0.195
BL 100kL10	1.464 ± 0.032	1.463 ± 0.044	1.472 ± 0.024	1.477 ± 0.023	1.492 ± 0.021	1.476 ± 0.040
BL 1ML10	2.733 ± 0.052	2.750 ± 0.076	2.797 ± 0.058	2.808 ± 0.071	2.822 ± 0.074	2.858 ± 0.056
BL 400kL25	1.760 ± 0.082	1.814 ± 0.113	1.854 ± 0.106	1.865 ± 0.099	1.887 ± 0.083	1.879 ± 0.107
BL 100k	1.374 ± 0.034	1.388 ± 0.031	1.390 ± 0.042	1.399 ± 0.027	1.404 ± 0.034	1.403 ± 0.046
BL 1ML25	2.306 ± 0.166	2.359 ± 0.178	2.412 ± 0.207	2.520 ± 0.155	2.427 ± 0.246	2.521 ± 0.220
BL 400kL10	2.143 ± 0.056	2.139 ± 0.039	2.191 ± 0.040	2.172 ± 0.045	2.199 ± 0.040	2.213 ± 0.039
RNDL 1ML20B	2.628 ± 0.179	2.844 ± 0.100	2.924 ± 0.066	2.936 ± 0.050	2.905 ± 0.058	2.982 ± 0.072
RNDL 400kL20	2.185 ± 0.081	2.138 ± 0.045	2.221 ± 0.033	2.225 ± 0.039	2.222 ± 0.041	2.227 ± 0.045
RNDL 400kL20B	2.170 ± 0.076	2.160 ± 0.043	2.211 ± 0.047	2.209 ± 0.026	2.185 ± 0.078	2.234 ± 0.040
RNDL 100kL20	1.443 ± 0.037	1.442 ± 0.056	1.433 ± 0.040	1.466 ± 0.021	1.453 ± 0.033	1.456 ± 0.038
RNDL 1ML20	2.630 ± 0.150	2.845 ± 0.084	2.845 ± 0.239	2.919 ± 0.054	2.909 ± 0.068	2.993 ± 0.108
4G 700k	2.198 ± 0.057	2.222 ± 0.045	2.229 ± 0.023	2.238 ± 0.042	2.218 ± 0.056	2.219 ± 0.044
4G 500k	2.070 ± 0.059	2.092 ± 0.059	2.092 ± 0.049	2.091 ± 0.059	2.081 ± 0.052	2.085 ± 0.042
4G 300k	1.628 ± 0.025	1.632 ± 0.046	1.660 ± 0.036	1.629 ± 0.033	1.629 ± 0.039	1.658 ± 0.029
FB 8M	4.295 ± 0.022	4.293 ± 0.017	4.297 ± 0.021	4.289 ± 0.019	4.289 ± 0.020	4.304 ± 0.019
FB 4M	4.286 ± 0.029	4.264 ± 0.023	4.283 ± 0.024	4.265 ± 0.020	4.266 ± 0.024	4.280 ± 0.022
FB 5M	4.289 ± 0.024	4.284 ± 0.022	4.284 ± 0.026	4.277 ± 0.023	4.275 ± 0.019	4.284 ± 0.034
FB 3M	4.246 ± 0.027	4.219 ± 0.020	4.258 ± 0.031	4.242 ± 0.023	4.241 ± 0.029	4.238 ± 0.028
FB 1M	3.251 ± 0.133	3.188 ± 0.034	3.270 ± 0.033	3.250 ± 0.035	3.241 ± 0.062	3.372 ± 0.044
FB 800k	2.962 ± 0.041	2.912 ± 0.063	3.061 ± 0.041	3.064 ± 0.026	3.021 ± 0.051	3.059 ± 0.040
FB 500k	2.501 ± 0.054	2.516 ± 0.036	2.539 ± 0.022	2.540 ± 0.035	2.535 ± 0.028	2.578 ± 0.067
FB 340k	2.360 ± 0.087	2.358 ± 0.035	2.388 ± 0.039	2.381 ± 0.027	2.376 ± 0.030	2.383 ± 0.032
FB 200k	2.133 ± 0.059	2.082 ± 0.081	2.148 ± 0.074	2.117 ± 0.027	2.120 ± 0.070	2.235 ± 0.043
FB 150k	1.749 ± 0.046	1.732 ± 0.027	1.794 ± 0.037	1.788 ± 0.049	1.751 ± 0.046	1.789 ± 0.050
FB 100k	1.534 ± 0.030	1.527 ± 0.048	1.537 ± 0.034	1.546 ± 0.027	1.569 ± 0.023	1.563 ± 0.037

Table 2: Average QoE rewards per network profile. Our proposed DIQL algorithm demonstrates robust performance across the majority of network profiles, outperforming the behavior policy as well as prior offline policy training algorithms.

Environment	IQL	WIQL
Halfcheetah Medium Expert	93.4 ± 0.6	93.3 ± 0.5
Halfcheetah Medium Replay	44.9 ± 0.2	44.5 ± 0.2
Halfcheetah Medium	48.0 ± 0.1	47.6 ± 0.1
Hopper Medium Expert	111.9 ± 0.7	111.6 ± 0.4
Hopper Medium Replay	99.1 ± 2.0	97.8 ± 1.4
Hopper Medium	64.9 ± 3.1	65.9 ± 4.4
Walker2d Medium Expert	112.6 ± 0.2	112.7 ± 0.4
Walker2d Medium Replay	82.9 ± 0.5	86.3 ± 3.4
Walker2d Medium	80.5 ± 0.6	83.0 ± 0.8

Table 3: D4RL MuJoCo benchmark: best normalized score.

the best D4RL-normalized score across seeds. This benchmark isolates the learning algorithm from the end-to-end RTC system. It can be observed based on Table 3 that the proposed algorithm matches or surpasses IQL across tasks, indicating that the distributional offline RL design is competitive beyond the RTC domain.

5 Conclusion

We presented a human-in-the-loop, data-driven framework for learning bandwidth estimators in RTC, combining QoE-aligned reward modeling with a distributional offline RL algorithm. Trained on roughly 1M Microsoft Teams call traces and deployed in the production media stack, the learned bandwidth estimator runs every 60ms with sub-millisecond inference time and a compact memory footprint. A two-week A/B test shows an 11.41% reduction in subjective poor call rate relative to the baseline estimator, alongside statistically significant gains in objective video quality scores and a negligible change in audio quality. Three elements proved decisive in practice: (i) coupling subjective protocols (P.808/P.910) with objective reward modeling so that optimization targets reflect user-perceived quality; (ii) using offline RL to learn from real data while avoiding risky on-line exploration; and (iii) exporting an ONNX model that meets latency targets, deploying it via a staged rollout, and monitoring QoE-related A/B deltas. Beyond bandwidth estimation, the learning algorithm was evaluated on the D4RL continuous-control benchmark, and showed method-level competitiveness independent of RTC integration.

References

- Agarwal, N.; Pan, R.; Yan, F. Y.; and Netravali, R. 2025. Mowgli: Passively Learned Rate Control for {Real-Time} Video. In *22nd USENIX Symposium on Networked Systems Design and Implementation (NSDI 25)*, 579–594.
- Ashvin, N.; Murtaza, D.; Abhishek, G.; and Sergey, L. 2020. Accelerating online reinforcement learning with offline datasets. *CoRR*, vol. *abs/2006.09359*.
- Bellemare, M. G.; Dabney, W.; and Munos, R. 2017. A distributional perspective on reinforcement learning. In *International conference on machine learning*, 449–458. PMLR.
- Bentaleb, A.; Akcay, M. N.; Lim, M.; Begen, A. C.; and Zimmermann, R. 2022. BoB: Bandwidth prediction for real-time communications using heuristic and reinforcement learning. *IEEE Transactions on Multimedia*.
- Bergkvist, A.; Burnett, D.; Jennings, C.; and Narayanan, A. 2012. Webrtc 1.0: Real-time communication between browsers. w3c working draft. *World Wide Web Consortium*.
- Carlucci, G.; De Cicco, L.; Holmer, S.; and Mascolo, S. 2016. Analysis and design of the google congestion control for web real-time communication (WebRTC). In *Proceedings of the 7th International Conference on Multimedia Systems, MMSys '16*. New York, NY, USA: Association for Computing Machinery. ISBN 9781450342971.
- Cetinkaya, E.; Pehlivanoglu, A.; U. Ayten, I.; Yumakogullari, B.; E. Ozgun, M.; K. Erinc, Y.; Deniz, E.; and C. Begen, A. 2024. Offline Reinforcement Learning for Bandwidth Estimation in RTC Using a Fast Actor and not-So-Furious Critic. In *Proceedings of the 15th ACM Multimedia Systems Conference*.
- Cuturi, M. 2013. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26.
- Dabney, W.; Ostrovski, G.; Silver, D.; and Munos, R. 2018a. Implicit quantile networks for distributional reinforcement learning. In *International conference on machine learning*, 1096–1105. PMLR.
- Dabney, W.; Rowland, M.; Bellemare, M.; and Munos, R. 2018b. Distributional reinforcement learning with quantile regression. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.
- Delon, J.; and Desolneux, A. 2020. A Wasserstein-type distance in the space of Gaussian mixture models. *SIAM Journal on Imaging Sciences*, 13(2): 936–970.
- Engelke, U.; and Zepernick, H.-J. 2007. Perceptual-based quality metrics for image and video services: A survey. In *2007 Next Generation Internet Networks*, 190–197. IEEE.
- Eo, J.; Niu, Z.; Cheng, W.; Yan, F. Y.; Gao, R.; Kardhashi, J.; Inglis, S.; Revow, M.; Chun, B.-G.; Cheng, P.; et al. 2022. OpenNetLab: Open platform for RL-based congestion control for real-time communications. *Proc. of APNet*.
- Fang, J.; Ellis, M.; Li, B.; Liu, S.; Hosseinkashi, Y.; Revow, M.; Sadovnikov, A.; Liu, Z.; Cheng, P.; Ashok, S.; et al. 2019. Reinforcement learning for bandwidth estimation and congestion control in real-time communications. *arXiv preprint arXiv:1912.02222*.
- Fu, J.; Kumar, A.; Nachum, O.; Tucker, G.; and Levine, S. 2020. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*.
- Fujimoto, S.; and Gu, S. S. 2021a. A minimalist approach to offline reinforcement learning. *Advances in neural information processing systems*, 34: 20132–20145.
- Fujimoto, S.; and Gu, S. S. 2021b. A minimalist approach to offline reinforcement learning. *Advances in neural information processing systems*, 34: 20132–20145.
- Gottipati, A.; Khairy, S.; Hosseinkashi, Y.; Mittag, G.; Gopal, V.; Yan, F. Y.; and Cutler, R. 2024. Balancing Generalization and Specialization: Offline Metalearning for Bandwidth Estimation. *arXiv preprint arXiv:2409.19867*.
- Gottipati, A.; Khairy, S.; Mittag, G.; Gopal, V.; and Cutler, R. 2023. Offline to Online Learning for Real-Time Bandwidth Estimation. *arXiv preprint arXiv:2309.13481*.
- Haarnoja, T.; Zhou, A.; Abbeel, P.; and Levine, S. 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, 1861–1870. Pmlr.
- Johansson, I.; et al. 2024. Self-Clocked Rate Adaptation for Multimedia (SCReAMv2) — IETF Internet-Draft. IETF Internet-Draft. Obsoletes RFC 8298.
- Khairy, S.; Mittag, G.; Gopal, V.; Yan, F. Y.; Niu, Z.; Ameri, E.; Inglis, S.; Golestaneh, M.; and Cutler, R. 2024. ACM MMSys 2024 Bandwidth Estimation in Real Time Communications Challenge. In *Proceedings of the 15th ACM Multimedia Systems Conference, MMSys '24*, 339–345. New York, NY, USA: Association for Computing Machinery. ISBN 9798400704123.
- Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*.
- Kostrikov, I.; Nair, A.; and Levine, S. 2021. Offline Reinforcement Learning with Implicit Q-Learning. In *Deep RL Workshop NeurIPS 2021*.
- Kozakowski, P.; Kaiser, L.; Michalewski, H.; Mohiuddin, A.; and Kańska, K. 2022. Q-value weighted regression: Reinforcement learning with limited data. In *2022 International Joint Conference on Neural Networks (IJCNN)*, 1–8. IEEE.
- Li, H.; Lu, B.; Xu, J.; Song, L.; Zhang, W.; Li, L.; and Yin, Y. 2022. Reinforcement learning based cross-layer congestion control for real-time communication. In *2022 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, 01–06. IEEE.
- Li, Z.; Aaron, A.; Katsavounidis, I.; Moorthy, A.; and Manohara, M. 2016. Toward a practical perceptual video quality metric. Netflix.
- Lillicrap, T. P.; Hunt, J. J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; and Wierstra, D. 2015. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*.
- Lu, B.; Wang, K.; Xu, J.; Song, L.; Xie, R.; and Zhang, W. 2024. Pioneer: Offline Reinforcement Learning based Bandwidth Estimation for Real-Time Communication. In *Proceedings of the 15th ACM Multimedia Systems Conference*.

Markudova, D.; and Meo, M. 2023. ReCoCo: Reinforcement learning-based Congestion control for Real-time applications. In *2023 IEEE 24th International Conference on High Performance Switching and Routing (HPSR)*, 68–74. IEEE.

Mittag, G.; Naderi, B.; Gopal, V.; and Cutler, R. 2023. LSTM-Based Video Quality Prediction Accounting for Temporal Distortions in Videoconferencing Calls. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1–5. IEEE.

Peng, X. B.; Kumar, A.; Zhang, G.; and Levine, S. 2019. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Schulzrinne, H.; Casner, S.; Frederick, R.; and Jacobson, V. 2003. RTP: A transport protocol for real-time applications. Technical report.

Strauss, J.; Katabi, D.; and Kaashoek, F. 2003. A measurement study of available bandwidth estimation tools. In *Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement*, 39–44.

Wang, B.; Zhang, Y.; Qian, S.; Pan, Z.; and Xie, Y. 2021. A hybrid receiver-side congestion control scheme for web real-time communication. In *Proceedings of the 12th ACM Multimedia Systems Conference*, 332–338.

Zhang, H.; Zhou, A.; Lu, J.; Ma, R.; Hu, Y.; Li, C.; Zhang, X.; Ma, H.; and Chen, X. 2020. OnRL: improving mobile video telephony via online reinforcement learning. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, 1–14.

Zhang, W.; Tao, X.; and Wang, J. 2024. NAORL: Network Feature Aware Offline Reinforcement Learning for Real Time Bandwidth Estimation. In *Proceedings of the 15th ACM Multimedia Systems Conference, MMSys '24*, 326–331. New York, NY, USA: Association for Computing Machinery. ISBN 9798400704123.

Zhu, X.; Pan, R.; Ramalho, M.; Cruz, R.; et al. 2020. RFC 8698: Network-Assisted Dynamic Adaptation (NADA). IETF RFC.