Pareto-optimized Content-hierarchical Adaptive Reinforcement Learning for QoE-driven Short-form Video Streaming

Pham Nhat Quang[1], Nguyen Cong Binh[2], Le Mai Nam[2], and Nguyen Viet Hung[2]

[1]*Vietnam National University, Hanoi, Vietnam,*
[2]*East Asia University of Technology, Hanoi, Vietnam*

https://doi.org/10.26636/jtit.2026.4.2798

Submitted: 11 Aug. 2026, Reviewed: 23 Sep. 2026, Accepted: 24 Sep. 2026

Abstract — The swipe-driven structure of short-form video creates a coupled control problem. The client must protect the playback of the current clip while deciding whether scarce bandwidth should be allocated to a subsequent clip that may not be viewed at all. Most adaptive bitrate (ABR) learners rely on a fixed scalar quality of experience (QoE) function, treating nominal bitrate as the primary proxy for the content value. This study presents a two-stage controller that separates feed-level prefetch allocation from chunk-level quality selection. Stage 1 converts early-swipe risk and playback state into a bounded prefetch budget. Stage 2 retains quality, rebuffering, smoothness, and wasted traffic as a four-dimensional reward and learns a preference-conditioned envelope Q-function. A content-aware utility ladder exposes the estimated value and measured byte cost of each candidate representation. A feasibility mask and a hysteresis filter constrain execution. Across four recorded network conditions, the controller achieves the highest official score among four evaluated systems: a four-trace mean QoE of 60.970 points and a 2.269 MB lower mean data wastage than the MPC baseline, which is equivalent to a 37.7% reduction. Under equal weight and tracewise normalization, its mean normalized streaming index is 0.980. A separate held-out evaluation of ten paired environment seeds reports a mean official score of 15.196 (95% CI: 11.324 – 19.068), while a 35-preference analysis yields a four-objective hypervolume of 0.433 (95% CI: 0.370 – 0.497) for the frozen stage 2 policy. These results highlight the potential of hierarchical reinforcement learning to improve user experience in swipe-driven video platforms.

Keywords — adaptive bitrate streaming, content-aware streaming, hierarchical reinforcement learning, Pareto optimization, quality of experience, short-form video streaming

1. Introduction

Short-form video has transformed streaming from long, mostly linear sessions into sequences of brief, weakly predictable viewing opportunities. A user may watch a clip until it ends, leave after a few seconds, or swipe repeatedly until a relevant item appears. The player must nevertheless make download decisions before that behavior is known. Therefore, it serves two demands over the same network path: keeping the current clip playable and preparing the next clip quickly enough that a swipe does not expose a blank screen. Downloading too little increases the start delay. Downloading too much into the feed consumes bandwidth for content that is never shown. Recent short video streaming systems treat swipe behavior, target selection, preloading, and bitrate adaptation as coupled control problems rather than as independent player modules [1]–[4].

Conventional ABR algorithms select one representation for each upcoming chunk from a fixed bitrate ladder. Throughput-driven rules react quickly, but inherit prediction error, whereas buffer-driven methods trade aggressiveness for a larger safety margin [5]. Model predictive control (MPC) explicitly combines a forecast with buffer evolution over a finite horizon [6]; Bola, meanwhile, casts the adaptation as online utility maximization driven primarily by buffer occupancy [7]. Recent QoE management surveys emphasize that ABR quality also depends on representation design, client observations, and network assistance [8]. These controllers are efficient and interpretable, but their objectives are typically fixed prior to deployment. In a short video feed, the relative cost of an additional byte changes with the probability that its corresponding chunk will be watched. A fixed scalar objective is unlikely to remain appropriate under startup, stable playback, and imminent swipe conditions.

Reinforcement learning (RL) offers a more flexible mapping from player observations to bitrate-selection actions: The Pensieve system learns an ABR rule directly from experienced QoE [9], later work optimizes QoE under bandwidth constraints [10], abrupt network variation [11], and recent neural-ABR work stresses trace realism and repeatable evaluation [12]. Short video systems have since explored multi-agent separation of target and bitrate decisions and learning-based preloading [2], [4], [10], [13]. However, most existing systems still reduce quality, rebuffering, smoothness, and bandwidth consumption to a single weighted sum. Changing the weights modifies the task and can require a new policy. Furthermore, a fixed bitrate ladder assigns the same marginal value to every title, regardless of content; Recent content-optimized ladder studies show that the complexity of the source actually changes the useful quality-bitrate operating region [14].

This work is licensed under a Creative Commons Attribution 4.0 International (CC BY 4.0) License.
For more information, see https://creativecommons.org/licenses/by/4.0/

Pareto-optimized content-hierarchical adaptive reinforcement learning (POCHA-RL) addresses these limitations by combining three ideas in one decision loop. First, a content-aware utility ladder maps each video to a small set of utility increments and retains the measured chunk size associated with every level.

Second, a hierarchical controller separates the feed-timescale question of *where to spend bandwidth* from the chunk-timescale question of *which level of quality to request*. Third, the low-level controller learns a vector-valued action function conditioned on an explicit preference vector. The preference may be sampled during training to cover the simplex and scheduled from buffer, network variability, content complexity, and skip probability at inference time. Hard constraints are applied before the learned ranking, and a small hysteresis rule stabilizes the final action. The learned model remains the action ranking mechanism. The surrounding rules define the operating envelope whose contribution must be isolated by ablation.

This study contributes to the following:

- a hierarchical multi-objective formulation for short-form ABR in which prefetch allocation and bitrate selection exchange a bounded resource signal,
- a content-aware state representation that couples the utility, candidate size, retention risk, buffer state, and throughput profile,
- a preference-conditioned envelope double DQN that separates vector reward measurement from action scalarization,
- a simulator-aligned, controller-level benchmark over four network conditions, seven retention curves, and three repository baselines.

The remainder of the paper follows the same order as is applied to the method itself. Section 2 reviews ABR, learning-based streaming, hierarchical control, and multiobjective RL. Section 3 specifies the architecture, state, reward, and learning rule. Section 4 describes the experimental configuration and metrics. Section 5 reports the measurements, normalization, and limitations, while Section 6 concludes the paper.

2. Related Works

2.1. Adaptive Bitrate Streaming

Classical ABR methods select chunk quality from throughput, buffer occupancy, or a finite-horizon model. Buffer-based adaptation and Bola show that the buffer state is capable of regularizing quality decisions under variable capacity [5], [7], while MPC makes the quality stall trade-off explicit by evaluating candidate sequences over a receding horizon [6]. These controllers are effective for mostly linear playback, but short-form feeds add a second resource question: whether the next byte should protect the current video or prepare a future one. Dashlet, predictive segment preloading, and learning-based preloading address this feed-aware pressure from different angles [1], [4], [15]. POCHA-RL adopts the same premise and adds content utility, retention risk, and byte cost to a preference-conditioned controller. Traffic wastage-aware ABR reinforces the need to model downloaded but unwatched data explicitly [16], while recent surveys identify representation design, observability, and network volatility as persistent QoE challenges [8].

2.2. Reinforcement Learning for Video Streaming

Learning-based ABR replaces handwritten thresholds with a policy trained on simulated player-network interactions. The Pensieve system learns to adapt the bitrate from throughput history, buffer state, previous bitrate, remaining chunks, and candidate sizes under a scalar QoE reward [9]. Related resilient adaptive streaming systems for 360-degree video show how buffer and bandwidth allocation can protect QoE under sudden capacity drops [11], [13].

More recent neural ABR work stresses practical evaluation and deployment stability [12] and double DQN addresses value overestimation by decoupling action selection from target evaluation [17]. Short video systems such as Incendio and Gamora add target selection, preloading, or expert guidance, while content-optimized ladder construction confirms that video complexity changes efficient representation points [2], [4], [14].

These works motivate learned control but still commonly bind deployment to one-scalar utility. Instead, the proposed method keeps four reward dimensions until preference-dependent ranking, while placing a content-derived utility proxy on both the offline ladder and the online candidate state.

2.3. Hierarchical Reinforcement Learning

Hierarchical RL decomposes long-horizon decisions into temporally distinct subproblems, as formalized by the options framework [18]. Short-form streaming has a natural hierarchy: feed-level allocation changes when the active video or retention context changes, whereas chunk quality is selected for every request. Therefore, POCHA-RL freezes a stage 1 skip estimator and exposes it to stage 2 as a soft prefetch budget and feasibility signal. The stages are not independent at runtime; they remain coupled through buffer state, watch probability, and the valid action set. Decomposition is a practical bias that aligns the learning frequency with the physical timescale of each decision.

Relative to representative ABR and short-video controllers, the proposed framework is positioned by the combination rather than the isolated presence of its ingredients: MPC, Bola, Pensieve, and recent neural-ABR systems mainly operate at the chunk level with a fixed scalar objective; Dashlet, Incendio, and Gamora incorporate feed-aware short-video structure, while POCHA-RL couples feed-level allocation, content-derived candidate utility, and preference-conditioned vector returns in one guarded execution loop. The literature comparison defines the architectural gap. The evaluation covers the Bola-basic approach alongside the three originally evaluated baselines.

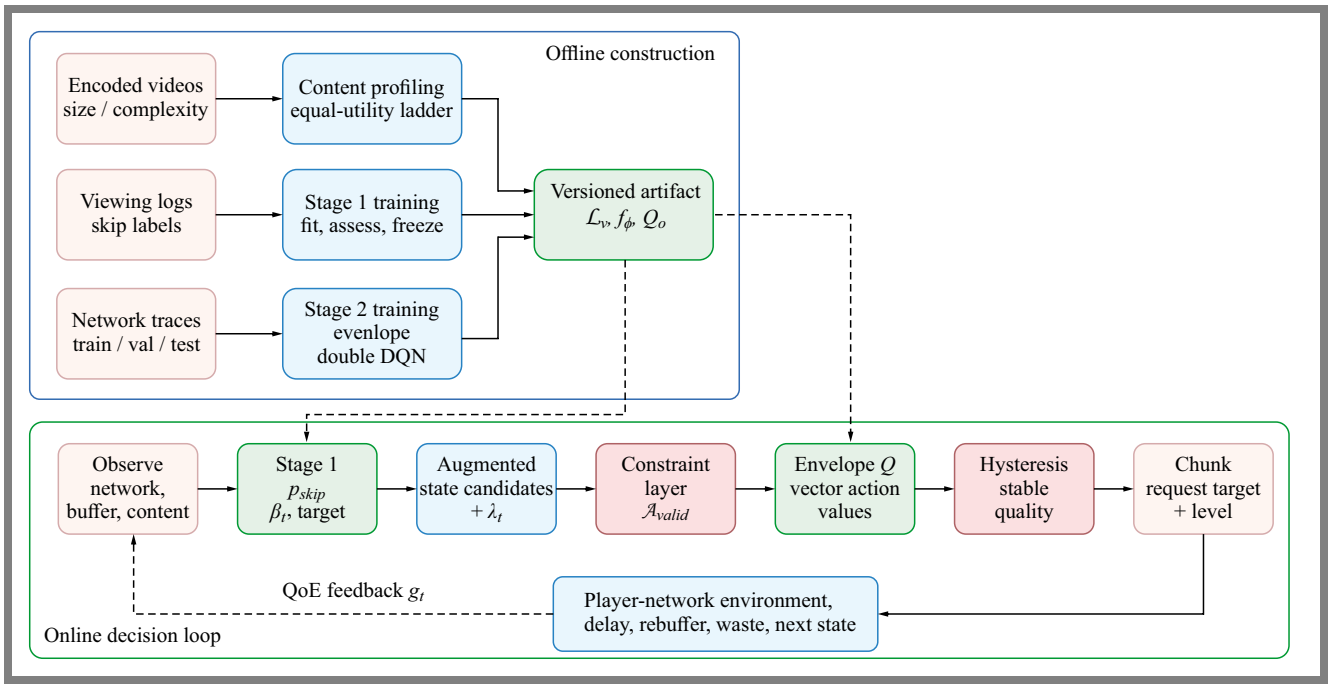


Fig. 1. Proposed architecture. Off-line processing produces three versioned artifacts. Online control applies stage 1 allocation, a feasibility mask, Q-ranking conditioned on preference, and output stabilization in a fixed order. The dashed arrows denote model injection and feedback rather than download flow.

2.4. Multi-objective and Pareto Optimization

Multiobjective sequential decisions are difficult when the objective importance changes after training: a fixed weighted sum identifies one operating point and may miss parts of the attainable frontier, a limitation emphasized in recent MORL taxonomies [19].

Preference-conditioned MORL supplies a weight vector to the value function, and envelope Q-learning enables one model to represent returns for many linear preferences [20].

This distinction is operationally useful in streaming because low buffer should emphasize rebuffer avoidance, stable playback can spend more on quality, and high skip probability should increase waste avoidance. POCHA-RL encodes these priorities explicitly while keeping hard safety constraints outside the Pareto comparison: the mask first defines feasible actions, and the preference-conditioned Q-network ranks only the survivors.

3. Proposed Framework

3.1. System Architecture

The architecture is divided into an *offline construction path* and an *online control path*. The former produces immutable artifacts from videos, behavior logs, and network traces, such as content ladders, a frozen skip predictor, and an envelope Q checkpoint. The latter consumes only observations available at decision time. This prevents test information from entering the controller and allows every online action to be traced to a versioned model and metadata record (Fig. 1).

Figure 1 also clarifies the role of the constraint layer. Retention, network profiling, and buffer thresholds shape the state and feasible set, but do not directly replace the learned bitrate ranking. This boundary is important experimentally. Ablation can remove the constraint or stabilization layer without changing the interface of the Q-network.

Consider a feed of videos indexed by v. Video v is divided into one-second chunks indexed by t and encoded at K quality levels. $b_{v,t}^{(k)}$, $S_{v,t}^{(k)}$, and $U_{v,t}^{(k)}$ denote bitrate, byte size, and normalized content-utility proxy of level $k \in \{1, \ldots, K\}$.

The player maintains a playback buffer B_t for the current video and may hold prefetched data for the adjacent video. At each download opportunity, it first chooses a target offset $o_t \in \{0, 1, \varnothing\}$, representing the current video, the next video, or a short sleep period. It then chooses a level $a_t \in \{1, \ldots, K\}$ for the target chunk.

For a requested chunk of duration T_{chunk} and download time d_t, the current-video buffer follows:

$$\begin{aligned} B_{t+1} &= \left[B_t - d_t\right]^+ + T_{\text{chunk}} \mathbb{I}[o_t = 0] \\ d_{\text{rb},t} &= \left[d_t - B_t\right]^+ \end{aligned}, \tag{1}$$

where $[x]^+ = \max(x, 0)$.

A download with $o_t = 1$ warms the next video queue but does not replenish the current buffer. If the user swipes before those bytes are rendered, they contribute to W_t. This asymmetric transition is the reason a target decision cannot be reduced to ordinary bitrate selection.

The control problem is a multi-objective Markov decision process with return

$$\boldsymbol{J}(\pi) = \mathbb{E}_{\pi}\left[\sum_{t=0}^{T-1} \gamma^{t} \boldsymbol{g}_{t}\right] . \tag{2}$$

A policy π_1 dominates π_2 if $J_i(\pi_1) \geqslant J_i(\pi_2)$ for every objective and the inequality is strict for at least one. The design objective is therefore a set of non-dominated policies or operating points rather than a single unconstrained maximizer.

The controller is organized into an off-line path and an on-line path. Offline processing derives a content-aware ladder, trains a skip probability model, freezes stage 1, and then trains a vector-valued Q-network over network traces. Online execution performs the following ordered operations:

$$\begin{aligned} &\text{observe} \rightarrow \text{budget/target} \rightarrow \text{mask} \\ &\quad \rightarrow \text{envelope Q} \rightarrow \text{stabilize} \rightarrow \text{download} . \end{aligned} \tag{3}$$

The order is important. Stage 1 controls the admissible use of resources at the feed timescale, while stage 2 chooses the quality of the chunk inside that envelope. The post-decision stabilizer may hold or limit an abrupt switch, but does not replace the learned ranking. Sequential training follows directly from this interface. If stage 1 mapping changed while stage 2 replay was collected, identical low-level states could correspond to different target availability and buffer transitions. Freezing stage 1 removes this moving interface without removing the runtime coupling.

The same bitrate can have different marginal values in videos, a motivation consistent with recent content-optimized ladder construction work [14]. Because decoded perceptual scores are not available in this experiment, we model this dependency with the saturating utility proxy

$$U(b, c) = 1 - e^{-\frac{\alpha_0 b}{1+c}} , \tag{4}$$

where $c \in [0, 1]$ is a content complexity score and α_0 controls the saturation rate.

Complexity is estimated from the ratio between high-quality and low-quality fragment sizes along with temporal size variation. The desired utilities are uniformly spaced

$$V_k = V_{\min} + \frac{k-1}{K-1}\left(V_{\max} - V_{\min}\right) , \tag{5}$$

and the corresponding bitrate is obtained by inverting Eq. (4).

When the simulator exposes a fixed set of representations, the desired levels are mapped to the nearest available encodes, and the measured chunk sizes are retained. Thus, the ladder provides a semantic action index, whereas $S_{v,t}^{(k)}$ exposes the actual network cost.

The saturation in Eq. (4) gives a smaller modeled return to top-level bitrate increments on low-complexity clips, while high-complexity clips require more bitrate to reach the same modeled utility. The agent therefore receives both $U_{v,t}^{(k)}$ and $S_{v,t}^{(k)}$ rather than a single bitrate index.

These values are deterministic model outputs, not measured VMAF or subjective scores. Replacing them with decoded perceptual measurements leaves the controller interface unchanged, but the present results demonstrate only that the controller is aware of a content-derived proxy, not that this proxy tracks perceptual quality.

Stage 1 estimates the probability that the user will skip the target video. Its compact feature vector contains progress within the current clip, content complexity, and a recent session skip rate. A small multilayer perceptron produces

$$p_{\text{skip},t} = \sigma\big(f_{\phi}(x_t)\big) . \tag{6}$$

More explicitly, $x_t = [w_t/D_v, c_v, \bar{s}_t]$ combines normalized watch progress, content complexity, and the recent skip fraction $\bar{s}_t$. With hidden state $h_t = \mathrm{ReLU}(W_1 x_t + b_1)$, the output is $\sigma(W_2 h_t + b_2)$. Parameters are trained with

$$\begin{aligned} \mathcal{L}_{\text{BCE}}(\phi) = -\frac{1}{N}\sum_{n=1}^{N}\big[&y_n \log p_n \\ &+ (1-y_n)\log(1-p_n)\big] . \end{aligned} \tag{7}$$

When an explicit retention curve $R_v(\tau)$ is available, the conditional probability of still watching h seconds ahead is

$$p_{\text{watch},t}(h) = \frac{R_v(\min\{t+h, L_v - 1\})}{\max\{R_v(t), 10^{-6}\}} . \tag{8}$$

Observed retention is preferred for horizon shaping. Equation (6) is the deployable fallback when such a curve is unavailable. The predictor is frozen prior to stage 2 training. Probability quality matters because binary accuracy measures thresholded separation, whereas the budget consumes the continuous value $p_{\text{skip},t}$.

The protocol therefore measures discrimination and expected calibration error on held-out sessions, following the distinction between accuracy and confidence calibration in neural predictors [21]. The measured calibration limitation is reported in Sections 4 and 5.8, rather than hidden by post-hoc relabeling.

A bounded soft budget converts the prediction into prefetch pressure:

$$\beta_t = \beta_{\max}(1 - p_{\text{skip},t}) \min\left(1, \frac{B_t}{B_{\text{target}}}\right) . \tag{9}$$

Equation (9) has a direct interpretation. Prefetch increases only when the next video is likely to be watched and the current buffer is capable of supporting it. In the implementation, the startup and reserve rules further set the next video actions to be invalid when the current buffer is too small or the throughput history is insufficient. Only the adjacent video is prefetched, which avoids an unbounded combinatorial target space.

Let $\widehat{d}_t^{(k)} = S_{v,t}^{(k)}/\widehat{C}_t$ be the predicted download time under conservative throughput $\widehat{C}_t$. The constraint layer constructs

$$\begin{aligned} \mathcal{A}_{\text{valid}}(s_t) = \{k : &\widehat{d}_t^{(k)} \leqslant B_t - m(o_t), \\ &|k - k_{t-1}| \leqslant \Delta_{\max}\} , \end{aligned} \tag{10}$$

where $m(o_t)$ is a target-dependent safety margin and $\Delta_{\max}$ limits extreme quality jumps.

During a cold start, the admissible quality is capped and next-video downloads are disabled until the current video reaches its reserve. If Eq. (10) produces an empty set, the controller selects the safest lowest-level action, or sleeps briefly. This deterministic fallback ensures that every run-time decision remains well defined.

3.2. Content-aware State Representation

Stage 2 observation is a concatenation of six semantically distinct groups and candidate features. It can be written as

$$s_t = [A_t \mid X_t \mid P_t \mid D_t \mid E_t \mid F_t \mid H_t] , \tag{11}$$

where the groups have the following roles:

- A_t (network/player) contains recent throughput and download time, current buffer, and the last selected level.
- X_t (content) contains complexity, remaining chunks, and the utility and normalized byte size of every quality candidate.
- P_t (cross-level control) contains target offset, normalized prefetch budget $\beta_t/\beta_{\max}$, and queue position.
- D_t contains the four-dimensional preference $\boldsymbol{\lambda}_t$ supplied to the value function.
- E_t (retention) contains $p_{\text{skip},t}$, conditional watch probability, and a startup complete indicator.
- F_t (network profile) contains the harmonic mean, coefficient of variation, trend, forecast error, and conservative capacity.
- $H_t = \{h_t^{(k)}\}_{k=1}^{K}$ contains candidate-specific download time, expected rebuffer, quality-switch cost, waste risk, and bandwidth cost.

For K actions, each candidate contributes a fixed feature block. This preserves the association between a level, its utility, and its byte cost. Values with open ranges are clipped after normalization and the same feature builder is used in training and inference. Preference $\boldsymbol{\lambda}_t$ is passed separately to the Q-network, so that changes in priority do not masquerade as changes in the physical environment.

For positive throughput observations $C_{t-n+1:t}$, the profile computes:

$$\begin{aligned} C_{\text{harm},t} &= \frac{n}{\sum_{i=0}^{n-1} \frac{1}{C_{t-i}}} , \\ \mathrm{CV}_t &= \frac{\mathrm{std}\,(C)}{\max(\mathrm{mean}(C), \varepsilon)} , \\ C_{\text{safe},t} &= \frac{C_{\text{harm},t}}{1 + 0.18\, e_t + 0.10\, \mathrm{CV}_t} , \end{aligned} \tag{12}$$

where e_t is the normalized one-step forecast error.

The harmonic mean reduces the influence of high isolated samples, while the last denominator introduces an explicit uncertainty discount. Together with buffer and skip risk, these signals form an interpretable inference-time preference prior.

An unnormalized schedule is:

$$\widetilde{\lambda}_{\text{rb}} = \epsilon + g_{\text{rb}} \sigma(1.65 - B_t + 1.25\, \mathrm{CV}_t) , \tag{13}$$

$$\widetilde{\lambda}_{\text{q}} = \epsilon + g_{\text{q}} \sigma(B_t - 2.2 - 1.1\, \mathrm{CV}_t + 0.55\, c) , \tag{14}$$

$$\widetilde{\lambda}_{\text{sm}} = \epsilon + g_{\text{sm}} (1 + 0.7\, \mathrm{CV}_t) , \tag{15}$$

$$\widetilde{\lambda}_{\text{w}} = \epsilon + g_{\text{w}} (1 + 2.2\, p_{\text{skip},t}) , \tag{16}$$

followed by $\lambda_i = \widetilde{\lambda}_i / \sum_j \widetilde{\lambda}_j$.

The numerical offsets and slopes in this schedule are deliberate calibration parameters, rather than learned constants. Buffers are measured in seconds, and the coefficient of variation is dimensionless. Hence, the value 1.65 places the rebuffering sigmoid close to a conservative reserve region, whereas 2.2 delays the quality priority sigmoid until the current-video buffer is comfortably above that reserve.

Volatility multipliers of 1.25 and 1.1, respectively, advance rebuffering priority and postpone quality priority under unstable throughput, 0.55 raises quality priority for more complex content, and the factor 2.2 on $p_{\text{skip},t}$ makes waste avoidance more influential when early abandonment is likely. These parameters were fixed as part of the controller calibration before the reported comparisons, and subsequent normalization ensures that they affect only the relative preference direction supplied to the vector Q-function.

The network therefore does not output a scalar QoE estimate. It predicts four discounted outcomes for each candidate, and the preference vector plus the feasibility mask is applied only at the final ranking step.

The schedule follows monotonic control requirements rather than fitting to the four test cells, with offsets aligned to the 1.25 s reserve and 3 s target buffer; the constants are not claimed globally optimal.

3.3. POCHA-RL Algorithm

The environment returns one reward component per design objective

$$\boldsymbol{g}_t = \left[g_t^{\text{q}}, g_t^{\text{rb}}, g_t^{\text{sm}}, g_t^{\text{w}} \right]^{\top} . \tag{17}$$

The experimental implementation calibrates these components in the physical units used by the evaluator. For target offset o_t and horizon $H(o_t)$, it uses:

$$g_t^{\text{q}} = \sum_{h=0}^{H(o_t)-1} p_{\text{quality},t}(h, o_t)\, \frac{b^{(a_t)}}{1000} , \tag{18}$$

$$g_t^{\text{rb}} = -1.85 \sum_{h=0}^{H(o_t)-1} d_{\text{rb},t+h} , \tag{19}$$

$$g_t^{\text{sm}} = -p_{\text{watch},t}(0)\, \frac{|b^{(a_t)} - b_t^{\text{ref}}|}{1000} , \tag{20}$$

$$g_t^{\text{w}} = -0.25 \sum_{h=0}^{H(o_t)-1} \left(1 - p_{\text{watch},t}(h)\right) S_{t+h,\text{MB}}^{(a_t)} , \tag{21}$$

where b is in kbps, d_{rb} in seconds, and S_{MB} in megabytes.

The factor $p_{\text{quality},t}$ discounts quality unlikely to be viewed, and b_t^{ref} is the stable bitrate. We use $H(0) = 3$ for the current

video and $H(1) = 1$ for prefetching. The vector form prevents a large quality term from numerically erasing a rare but severe stall.

For simulator-aligned horizon training and diagnostics, the implementation additionally forms:

$$u_t^+(\boldsymbol{\lambda}) = \boldsymbol{\lambda}^\top \boldsymbol{g}_t - 4 \sum_{h=0}^{H(o_t)-1} S_{t+h,\mathrm{MB}}^{(a_t)} + \phi_{B,t} \,, \qquad \phi_{B,t} = \begin{cases} +0.12 & B_t^{\mathrm{after}} \geqslant 1.25 \,, \\ -0.60(1.25 - B_t^{\mathrm{after}}) & B_t^{\mathrm{after}} < 1.25 \,, \end{cases} \tag{22}$$

with $\boldsymbol{\lambda} \geqslant \mathbf{0}$ and $\sum_i \lambda_i = 1$.

Factor 4 is the official score penalty per downloaded MB, and $\phi_{B,t}$ enforces the 1.25 s reserve. This is a training/diagnostic signal; online selection remains the Q ranking of the masked envelope in Eq. (24), not a one-step maximization of u_t^+.

Equations (21) and (22) define the controller-side signal. The official evaluator in Eqs. (29) and (30) independently recomputes the raw session outcomes. This separation exposes every conversion step, so a genuine reduction in expected waste must show up as fewer measured unviewed bytes.

$c_{rb} = 1.85$ matches the evaluator's QoE definition in Eq. (29). The factor 4 follows from the official traffic charge and is untuned, $c_w = 0.25$ avoids double-charging already penalized bytes and the 1.25 s reserve equals one chunk plus 0.25 s headroom.

The network takes $(s_t, \boldsymbol{\lambda}_t)$ and outputs a vector for every action,

$$Q_\theta(s_t, \boldsymbol{\lambda}_t) \in \mathbb{R}^{K \times 4} \,. \tag{23}$$

Its shared backbone contains fully connected layers with layer normalization and ReLU activations. During exploration, an action is sampled uniformly from $\mathcal{A}_{\text{valid}}$. Otherwise

$$a_t = \operatorname*{arg\,max}_{a \in \mathcal{A}_{\text{valid}}(s_t)} \boldsymbol{\lambda}_t^\top Q_\theta(s_t, a, \boldsymbol{\lambda}_t) \,. \tag{24}$$

For a replay transition $(s_t, \boldsymbol{\lambda}_t, a_t, \boldsymbol{g}_t, s_{t+1})$, double DQN first selects the next action using the online network

$$a^\star = \operatorname*{arg\,max}_{a \in \mathcal{A}_{\text{valid}}(s_{t+1})} \boldsymbol{\lambda}_{t+1}^\top Q_\theta(s_{t+1}, a, \boldsymbol{\lambda}_{t+1}) \,, \tag{25}$$

and evaluates it with target parameters θ^-

$$\boldsymbol{y}_t = \boldsymbol{g}_t + \gamma(1 - z_t)\, Q_{\theta^-}(s_{t+1}, a^\star, \boldsymbol{\lambda}_{t+1}) \,, \tag{26}$$

where z_t is the terminal indicator. For a minibatch $\mathcal{B}$, the vector loss is

$$\mathcal{L}(\theta) = \frac{1}{4|\mathcal{B}|} \sum_{i \in \mathcal{B}} \sum_{m=1}^{4} \mathrm{Huber}_\delta\big(y_{i,m} - Q_{\theta,m}(s_i, a_i, \boldsymbol{\lambda}_i)\big) \,. \tag{27}$$

Target parameters are periodically copied and gradients are clipped. Training preferences are drawn from a mixture of contextual preferences, balanced samples, and corner-biased Dirichlet samples. The replay sampler preserves all four corner regions and a balanced region, avoiding a buffer dominated by quality-oriented trajectories.

The action mask acts before Eq. (24) and hysteresis acts after it. Let q_t^{raw} be the Q-selected level and q_t^{ref} the reference level carried from the previous stable decision. Promotion is accepted only if the estimated download can be sustained while preserving the buffer reserve. A downgrade is accepted immediately under predicted rebuffering. Otherwise, the final decision is limited to a one-level movement

$$q_t^{\mathrm{final}} = \begin{cases} q_t^{\mathrm{raw}}, & \widehat{d}_t^{(q_t^{\mathrm{raw}})} \leqslant B_t - B_{\mathrm{reserve}} \,, \\ \max(0, q_t^{\mathrm{ref}} - 1), & d_{\mathrm{rb},t}^{\mathrm{pred}} > 0 \,, \\ q_t^{\mathrm{ref}} + \Delta q_t, & \text{otherwise} \,, \end{cases} \tag{28}$$

where $\Delta q_t = \mathrm{clip}(q_t^{\mathrm{raw}} - q_t^{\mathrm{ref}}, -1, 1)$.

This layer cannot promote an action removed by the feasibility mask. Logging both q_t^{raw} and q_t^{final} makes its contribution measurable and prevents the stabilizer from being mistaken for policy competence.

Training is decoupled, but executed with the same feature and mask builders as used at inference. First, each ladder $\mathcal{L}_v$ is built, f_ϕ is trained with Eq. (7), and ϕ is frozen. Stage 2 then samples traces, videos, retention curves, and preference regions, computes p_{skip}, β_t, s_t, H_t, and $\mathcal{A}_{\mathrm{valid}}$, selects by masked ϵ-greedy Eq. (24), stores $(s_t, \boldsymbol{\lambda}_t, a_t, \boldsymbol{g}_t, s_{t+1}, z_t)$, and updates Q_θ with Eqs. (26) and (27) using preference-stratified replay.

In inference, random sampling and gradient updates are disabled. The contextual scheduler supplies $\boldsymbol{\lambda}_t$, Eq. (10) removes unsafe actions, Eq. (24) produces the raw action, and Eq. (28) yields the executed level.

The asymptotic inference cost of the learned stage is one forward pass plus $O(K)$ scalar products. Candidate construction is also $O(K)$, and the content ladder is computed offline. This complexity statement does not establish client latency. The recorded controller running time field is reported separately in Section 5.

4. Experimental Setup

4.1. Experimental Configuration

The evaluation uses the repository simulator with one-second chunks and four active bitrate levels: 750, 1200, 1850, and 2700 kbps. Seven retention curves describe heterogeneous viewing behavior. Network inputs are grouped into high, mixed, medium, and low conditions. The measured throughput sequences retained in the evaluation benchmark verify that the groups differ in both level and short-term variability.

The evaluation uses a trace-driven simulation instead of a deployed end-to-end system. It does not include a DASH/HLS client, decoder, or device telemetry, so the recorded running time reflects the simulator execution rather than the latency of the player. A fixed 60/20/20 split with four held-out traces per class and five sessions per trace is evaluated over ten paired environment seeds, giving 800 sessions per controller. The controllers remain frozen throughout these runs and no parameters are fitted to the evaluation data.

The reference configuration uses ten recent throughput samples and five candidate future chunks. Stage 1 has $\beta_{\max} = 0.30$ and a 3 s target buffer. The envelope DQN uses a 128-unit

hidden width, a replay capacity of 200 000 transitions, a batch size of 256, discount factor $\gamma = 0.92$, and a target-network update every 1000 optimization steps. The exploration rate ranges from 1.0 to 0.05. These settings are fixed for all reported controller comparisons.

These settings were fixed before evaluation from temporal-coverage and compute bounds, not tuned per trace ($\gamma = 0.92$ gives an effective horizon of 12.5 decisions). They are design choices, not claimed optima, distinct from the sensitivity study below.

The evaluation benchmark supplies four measured throughput groups and seven retention curves. The high trace has the largest capacity with visible bursts, the mixed trace has comparable local peaks but stronger reversals, the medium trace remains moderate, and the low trace is consistently capacity-limited. The retention curves also span slow decay and early abandonment. These inputs justify evaluating the controller across the crossed network and behavior conditions rather than associating one retention profile with one trace class.

Stage 1 is evaluated separately because a thresholded classifier and a continuous resource signal serves different purposes. On the held-out split generated from the seven retention curves, the predictor obtains 0.8000 accuracy, 0.8813 ROC-AUC, 0.8639 binary F1 score, and 0.1489 expected calibration error. The test labels are 79.05% positive, so an always-positive classifier reaches 0.7905 accuracy and a binary F1 score of 0.8830. Because Eq. (9) consumes the continuous $p_{\mathrm{skip},t}$ rather than a thresholded label, ranking quality is more relevant to the interface than binary accuracy alone.

ROC-AUC indicates useful ordering, while class imbalance and calibration error prevent the probability from being treated as an oracle. Stage 1 is consequently used only as a bounded contextual signal; calibration and external behavioral validity remain explicit limitations [21].

4.2. Evaluation Metrics

For each watched chunk, the simulator computes

$$Q_t = \frac{b_t}{1000} - 1.85\, d_{\mathrm{rb},t} - \frac{|b_t - b_{t-1}|}{1000}\,, \tag{29}$$

where bitrate is in kbps and rebuffering is in seconds. Session QoE is $\sum_t Q_t$.

The official Score additionally charges all downloaded traffic

$$\mathrm{Score} = \sum_t Q_t - 4D_{\mathrm{MB}}\,, \tag{30}$$

where D_{MB} is the downloaded volume in megabytes.

QoE and the official score are higher-is-better metrics. Data wastage is the volume downloaded but not viewed by a swipe. The download/watch ratio compares all downloaded bytes with the bytes corresponding to the watched content. A value closer to one is more efficient. The evaluation benchmark's "running time" field is retained as recorded controller runtime. Because its instrumentation boundary is not decomposed into per-decision and tail latency, it is used only for relative comparison within the retained benchmark and not as a client-deployment latency claim.

For a session with watched set $\mathcal{T}_{\mathrm{view}}$, the auxiliary diagnostics are:

$$T_{\mathrm{rb}} = \sum_{t\in\mathcal{T}_{\mathrm{view}}} d_{\mathrm{rb},t}\,, \qquad P_{\mathrm{sm}} = \sum_{t\in\mathcal{T}_{\mathrm{view}}} |b_t - b_{t-1}|\,, \tag{31}$$

$$W_{\mathrm{MB}} = \frac{1}{10^6}\sum_t W_t\,, \qquad \rho_{\mathrm{dw}} = \frac{D_{\mathrm{downloaded}}}{D_{\mathrm{watched}}}\,. \tag{32}$$

Startup is separated from mid-session rebuffering because prefetch can improve one while worsening the other.

5. Results and Discussion

5.1. Overall Performance

All results are computed using the official evaluator presented in Eqs. (29) and (30). The training time signal u_t^+ defined earlier is never substituted for reported metrics. Measurements are first presented in their original units, ensuring that the normalized comparison described later in this section remains anchored to practical magnitudes.

Repeated-seed, sensitivity, and all-trace metrics use paired seed-level Student-t CIs and exact sign-flip tests, Holm-corrected (nine comparisons for repeated-seed, four for Bola). Pareto hypervolume uses a common normalization box and shared Monte Carlo probes.

As shown in Tab. 1, POCHA-RL achieves the highest official score under all network conditions: 22.322, 22.669, 0.211, and 3.783 points on the high, mixed, medium and low traces, respectively. The framework also achieves the highest QoE on the high, mixed, and medium traces. On the low trace, fixed-preload surpasses it by just 0.630 QoE points, yet POCHA-RL maintains a higher official score and lower data wastege. This distinction is important – maximizing rendered quality alone can reward aggressive downloading even when the additional bytes are not useful. The official score exposes that cost through Eq. (30).

Averaged across the four traces, the proposed algorithm reaches 60.970 QoE points, compared to 39.215 for MPC, 48.793 for fixed preload and 41.858 for NextOne. Its mean official score is 10.355 points, while the corresponding values are 29.595, 7.400, and –13.460 points. The mean waste is 3.750 MB for POCHA-RL and 6.019 MB for MPC, an absolute reduction of 2.269 MB, or 37.7%. The average download/watch ratio decreases from 1.263 for MPC to 1.234, a relative reduction of 2.3%.

Within this benchmark, the joint pattern is consistent with the design premise that retention and buffer context should constrain prefetching before bitrate is optimized. It does not isolate which module produced the difference.

5.2. Repeated-seed Statistical Robustness

Across ten pairs of seeds (Tab. 2), POCHA-RL attains a mean score of 15.196, QoE 79.072, and waste of 2.237 MB, with

Tab. 1. Raw measured performance under four network conditions.

Trace	Metric	**POCHA-RL**	MPC	Fixed-preload	NextOne
High	QoE	**91.652**	85.130	67.270	67.280
	Score	**22.322**	14.230	17.180	1.000
	Waste [MB]	5.540	6.150	**4.161**	8.615
	Download/watch	**1.247**	1.300	1.320	1.780
	Running time [s]	0.470	0.320	0.090	**0.080**
Mixed	QoE	**86.673**	82.750	70.320	67.570
	Score	**22.669**	14.320	22.000	10.070
	Waste [MB]	**3.985**	5.975	4.295	6.450
	Download/watch	**1.196**	1.250	1.270	1.500
	Running time [s]	0.330	0.300	0.090	**0.080**
Medium	QoE	**48.374**	30.600	39.770	46.840
	Score	**0.211**	–37.470	–5.130	–2.990
	Waste [MB]	**4.045**	5.975	4.295	4.564
	Download/watch	**1.247**	1.250	1.270	1.300
	Running time [s]	0.350	0.300	**0.080**	**0.080**
Low	QoE	17.180	–41.620	**17.810**	–14.260
	Score	**–3.783**	–109.460	–4.450	–61.920
	Waste [MB]	**1.432**	5.975	1.788	4.045
	Download/watch	**1.247**	1.250	1.320	1.250
	Running time [s]	0.300	0.300	**0.090**	**0.090**

advantages of the paired score of 94.887, 16.360, and 49.705 over the MPC family, Fixed-preload and NextOne, respectively (all 95% CI exclude zero, Holm-corrected $p_{\mathrm{adj}} = 0.0176$). These are inference-time repetitions of frozen controllers, not retraining seeds.

The gains are not uniform across all objectives. Fixed-preload wastes less data than proposed on the high trace, showing that the integrated controller does not dominate every metric. On the contrary, the largest benchmark-level differences from MPC occur under medium and low capacity, where the repository baseline accumulates severe QoE penalties. This pattern motivates a component-level test of the aggregate results constraint layer, but the retained aggregate outcomes alone cannot assign the improvement to that layer.

Figure 2 visualizes the same raw measurements to expose trace-wise response patterns across objectives. It reveals a property that is less obvious from the scalar raw measurements alone. POCHA-RL does not obtain its official score ranking simply by selecting the lowest quality. Its QoE remains highest on the first three traces, while its download/watch ratio stays close to one. By the definition given in Eq. (30), the observed official score pattern reflects the joint QoE and the volume outcomes rather than the reported quantity in isolation.

Tab. 2. Repeated seed results (mean ±SD; $n = 10$ paired seeds).

Controller	Score	QoE	Waste [MB]
POCHA-RL	15.20 ±5.41	79.07 ±11.79	2.24 ±0.28
MPC-family	–79.69 ±6.04	41.88 ±10.23	4.24 ±0.57
Fixed-preload	–1.16 ±6.10	65.59 ±12.19	3.95 ±0.40
NextOne	–34.51 ±3.04	59.08 ±10.66	5.33 ±0.34

5.3. Trace-wise Behavior and Efficiency

The recorded running-time field is highest for POCHA-RL. Its four-trace mean is 0.363 s, versus 0.305 s for the MPC and approximately 0.09 s for the two fixed policies. Because the evaluation benchmark does not decompose this field into model inference, feature construction, or per-decision tail latency, the values establish only relative runtime under the recorded setup. A deployment claim requires instrumented client-side timing with a documented boundary.

The high-trace result also exposes a useful failure mode. Fixed preload attains the lowest waste under that condition because its prefetch volume is conservative in construction. The proposed framework spends part of the abundant capacity on higher expected quality, which raises waste if the user swipes. However, in medium and low capacity, its reserve and retention constraints prevent the official score collapse observed for MPC and NextOne. Therefore, the remaining tuning problem is specific: preserve low-capacity protection while moving the high-capacity operating point toward the Fixed-preload waste level when λ_w is large.

5.4. Pareto-front and Preference-coverage Analysis

To quantify preference coverage of the frozen stage 2 checkpoint, we sweep 35 preference vectors on the four-objective simplex, evaluated over ten seeds on the mixed trace against a buffer-safe heuristic (500 000 Monte Carlo probes, common

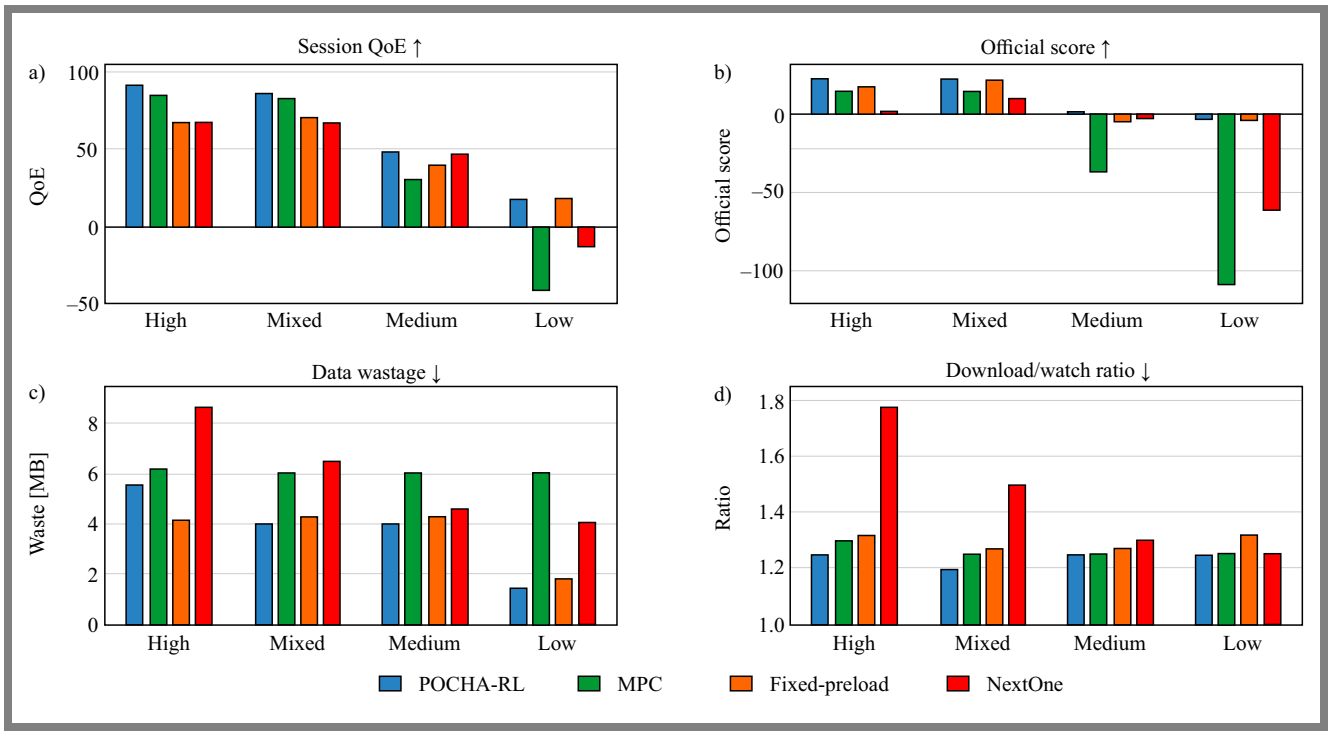


Fig. 2. Performance measured under network conditions. The panels expose the different shapes of quality response, official score, wasted traffic, and download efficiency.

normalization box). The preference-conditioned policy yields 32 distinct, fully nondominated outcomes versus one for the heuristic, with hypervolume 0.433 ±0.089 vs. 0.163 ±0.032 paired difference 0.270, $p = 0.00195$. Objective monotonicity is significant for quality, rebuffering, and smoothness $p < 0.001$ but not for waste $p = 0.617$, an open limitation.

5.5. Normalized Consolidated Results

To compare the metrics on a common scale without repeating another full table, the five measured metrics were normalized within each network condition. QoE and the official score were treated as benefit metrics, while waste, download/watch ratio, and run time were treated as cost metrics.

Table 3 reports the trace-wise normalized values. The equal weight normalized streaming index (NSI), computed on QoE, official score, waste, and download/watch ratio with runtime reported separately, gives POCHA-RL a four-trace mean of 0.980, compared with 0.692 for fixed preload, 0.428 for MPC, and 0.303 for NextOne.

The low-trace condition remains the only close exception in a single component: Fixed preload leads to normalized QoE by 0.011, but the proposed method leads to the other three streaming dimensions and reaches 0.997 NSI. Overall, normalized analysis reinforces the raw measured results rather than replacing them. The raw measured values remain the primary evidence for the practical magnitude.

5.6. Parameter Rationale and Sensitivity

We apply ±20% one-factor perturbations to c_{rb}, c_{w}, and the 1.25 s reserve on the frozen controller (traffic coefficient 4 untouched, as it redefines score itself, no retraining). The reserve is most consequential (–0.146 score at 1.00 s, –0.089 at 1.50 s, Tab. 4). Reward-coefficient effects are at most 0.040, all CIs containing zero, supporting only local, not global, robustness.

5.7. All-trace Bola Comparison and Scope Expansion

We compare frozen POCHA-RL against a published-parameter Bola-Basic adapter (current item only, $V = 1.754$ [7]) across all 80 traces (4000 sessions/controller, Tab. 5). Score improves by +42.703, QoE by +27.159, waste by –1.358 MB, and rebuffering by –13.909 s (all significant, $p_{\mathrm{adj}} = 0.00781$). No Pensieve row is reported [9], no compatible checkpoint exists, and a valid comparison requires independent retraining under the same split. This reuses the same seven video/retention profiles. The content and population diversity remain future work.

5.8. Limitations

Evidence should be interpreted within the scope of this evaluation artifact. The proposed framework is also evaluated as an integrated deployable controller. The feasibility mask, the hysteresis filter, stage 1 prefetch signal, and the vector Q network are executed together, so the available results cannot isolate which module drives the observed improvement. Behavioral validity is bounded by the seven retention curves. Stage 1 provides useful ranking evidence through ROC-AUC, but its calibration error and class imbalance mean that p_{skip} should be treated as a contextual signal rather than a population-level probability [21].

Tab. 3. Trace-wise min–max normalized results.

Trace	Controller	$\widehat{\text{QoE}}$	$\widehat{\text{Score}}$	$\widehat{\text{Waste}}$	$\widehat{\text{Ratio}}$	$\widehat{\text{Time}}$	NSI
High	**POCHA-RL**	**1.000**	**1.000**	0.691	**1.000**	0.000	**0.923**
	MPC	0.732	0.620	0.554	0.900	0.385	0.702
	Fixed-preload	0.000	0.759	**1.000**	0.863	0.974	0.655
	NextOne	0.000	0.000	0.000	0.000	**1.000**	0.000
Mixed	**POCHA-RL**	**1.000**	**1.000**	**1.000**	**1.000**	0.000	**1.000**
	MPC	0.795	0.337	0.193	0.822	0.120	0.537
	Fixed-preload	0.144	0.947	0.874	0.756	0.960	0.680
	NextOne	0.000	0.000	0.000	0.000	**1.000**	0.000
Medium	**POCHA-RL**	**1.000**	**1.000**	**1.000**	**1.000**	0.000	**1.000**
	MPC	0.000	0.000	0.000	0.939	0.185	0.235
	Fixed-preload	0.516	0.858	0.870	0.563	**1.000**	0.702
	NextOne	0.914	0.915	0.731	0.000	**1.000**	0.640
Low	**POCHA-RL**	0.989	**1.000**	**1.000**	**1.000**	0.000	**0.997**
	MPC	0.000	0.000	0.000	0.955	0.000	0.239
	Fixed-preload	**1.000**	0.994	0.922	0.000	**1.000**	0.729
	NextOne	0.460	0.450	0.425	0.955	**1.000**	0.573

Finally, the utility ladder is a deterministic surrogate rather than decoded VMAF or subjective quality, and normalized results depend on the selected baselines and equal weights of the NSI.

The held-out, Pareto, sensitivity, and all-trace experiments provide repeat-level confidence intervals, explicit Pareto-front and hypervolume analyses, local parameter robustness, and a paired comparison with Bola-Basic. These experiments do not establish training run stability, global hyperparameter optimality, or module-level attribution. Perceptual validity, real-player feasibility, and a protocol-matched Pensieve comparison remain subjects for future evaluation. Consequently, "content-aware" denotes the use of the stated surrogate rather than the validated perceptual quality.

6. Conclusions

This study has presented POCHA-RL, a hierarchical controller for short-form video streaming. The method estimates retention risk and allocates a bounded prefetch budget before selecting chunk quality with a content-aware, preference-conditioned vector Q-function. A utility ladder separates the modeled content value from the byte cost, while a feasibility mask and a hysteresis filter keeps the preference-conditioned ranking inside a buffer-safety envelope. This design preserves quality, buffering, smoothness, and wasted traffic as separate quantities until action selection.

Tab. 4. One-factor sensitivity ($\pm 20\%$); Δscore mean [95% CI] vs. nominal.

Parameter	Setting	Δscore [95% CI]
c_{rb}	$1.85 \rightarrow 1.48$	-0.040 $[-0.130, 0.050]$
c_{rb}	$1.85 \rightarrow 2.22$	-0.005 $[-0.100, 0.091]$
c_{w}	$0.25 \rightarrow 0.20$	-0.000 $[-0.003, 0.002]$
c_{w}	$0.25 \rightarrow 0.30$	-0.003 $[-0.008, 0.002]$
B_{reserve}	$1.25 \rightarrow 1.00$ s	-0.146 $[-0.291, -0.001]$
B_{reserve}	$1.25 \rightarrow 1.50$ s	-0.089 $[-0.235, 0.057]$

Tab. 5. Supplemental all-trace comparison. Values are mean $\pm$95% Student-t CI across ten class-balanced paired seed blocks; lower waste and rebuffering are better.

Controller	Score	QoE	Waste	Rebuf.
POCHA-RL	14.6 ±3.9	78.5 ±8.4	2.26 ±0.20	15.9 ±0.7
Bola-Basic	–28.1 ±3.3	51.3 ±8.4	3.62 ±0.32	29.8 ±1.6

Across the four network conditions, POCHA-RL achieves the highest official score under all network conditions, obtaining also favorable mean QoE, waste, and download efficiency relative to MPC, fixed-preload, and NextOne. The high trace still reveals a conservative waste advantage for Fixed-preload, and POCHA-RL incurs the highest recorded runtime. These are acknowledged as tuning and deployment trade-offs rather than being hidden by the normalized analysis.

The held-out evaluation provides repeat-level statistical evidence, while the Pareto, sensitivity, and all-trace analyses characterize preference coverage, parameter robustness, and performance across network traces. Perceptual and real-system validation remain topics for future evaluation.

References

[1] Z. Li, Y. Xie, R. Netravali, and K. Jamieson, "Dashlet: Taming Swipe Uncertainty for Robust Short Video Streaming", *arXiv*, 2023 (https://doi.org/10.48550/arXiv.2204.12954).

[2] Y. Li *et al.*, "Improving ABR Performance for Short Video Streaming Using Multi-agent Reinforcement Learning with Expert Guidance", *Proc. of the 33rd Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV '23)*, pp. 58–64, 2023 (https://doi.org/10.1145/3592473.3592564).

[3] V.H. Nguyen and T.H. Truong, "An Effective Segment Pre-fetching for Short-form Video Streaming", *International Journal of Computer Science and Network Security*, vol. 23, pp. 81–93, 2023 (https://doi.org/10.22937/IJCSNS.2023.23.3.8).

[4] B. Hou *et al.*, "Gamora: Learning-based Buffer-aware Preloading for Adaptive Short Video Streaming", *IEEE Transactions on Parallel and Distributed Systems*, vol. 35, pp. 2132–2146, 2024 (https://doi.org/10.1109/TPDS.2024.3456567).

[5] T.-Y. Huang *et al.*, "A Buffer-based Approach to Rate Adaptation: Evidence from a Large Video Streaming Service", *Proc. of ACM Conf. on Special Interest Group on Data Communication*, pp. 187–198, 2014 (https://doi.org/10.1145/2619239.2626296).

[6] X. Yin, A. Jindal, V. Sekar, and B. Sinopoli, "A Control-theoretic Approach for Dynamic Adaptive Video Streaming over HTTP", *Proc. of ACM Conference on Special Interest Group on Data Communication*, pp. 325–338, 2015 (https://doi.org/10.1145/2785956.2787486).

[7] K. Spiteri, R. Urgaonkar, and R.K. Sitaraman, "BOLA: Near-optimal Bitrate Adaptation for Online Videos", *IEEE/ACM Transactions on Networking*, vol. 28, pp. 1698–1711, 2020 (https://doi.org/10.1109/TNET.2020.2996964).

[8] R. Kalan and I. Dulger, "A Survey on QoE Management Schemes for HTTP Adaptive Video Streaming: Challenges, Solutions, and Opportunities", *IEEE Access*, vol. 12, pp. 170803–170839, 2024 (https://doi.org/10.1109/ACCESS.2024.3491613).

[9] H. Mao, R. Netravali, and M. Alizadeh, "Neural Adaptive Video Streaming with Pensieve", *Proc. of ACM Conference on Special Interest Group on Data Communication*, pp. 197–210, 2017 (https://doi.org/10.1145/3098822.3098843).

[10] P.D. Huynh and N.V. Hung, "Vega-360: Viewport-aware Hierarchical Grouped Allocation for Multi-layer 360° Video Streaming", *Jordanian Journal of Computers & Information Technology*, vol. 12, pp. 52–66, 2026 (https://doi.org/10.5455/jjcit.71-1764146706).

[11] V.H. Nguyen *et al.*, "Scalable and Resilient 360-degree-video Adaptive Streaming over HTTP/2 Against Sudden Network Drops", *Computer Communications*, vol. 216, pp. 1–15, 2024 (https://doi.org/10.1016/j.comcom.2024.01.001).

[12] S. Patel, J. Zhang, N. Narodystka, and S.A. Jyothi, "Practically High Performant Neural Adaptive Video Streaming", *Proc. of the ACM on Networking*, vol. 2, pp. 1–23, 2024 (https://doi.org/10.1145/3696401).

[13] N.V. Hung *et al.*, "Heverl-Viewport Estimation Using Reinforcement Learning for 360-degree Video Streaming", *Informatics and Automation*, vol. 24, pp. 302–328, 2025 (https://doi.org/10.15622/ia.24.1.11).

[14] A. Telili *et al.*, "Convex Hull Prediction Methods for Bitrate Ladder Construction: Design, Evaluation, and Comparison", *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 21, pp. 1–23, 2025 (https://doi.org/10.1145/3723006).

[15] H.L. Nguyen, T.H. Truong, V.H. Nguyen, and N.N. Pham, "Efficient Short-form Video Streaming: An Integration of Dynamic Bitrate Adaptation and Predictive Segment Preloading", *15th International Conference on Ubiquitous and Future Networks (ICUFN)*, Budapest, Hungary, 2024 (https://doi.org/10.1109/ICUFN61752.2024.10625216).

[16] H. Su *et al.*, "Reducing Traffic Wastage in Video Streaming via Bandwidth-efficient Bitrate Adaptation", *IEEE Transactions on Mobile Computing*, vol. 23, pp. 10361–10377, 2024 (https://doi.org/10.1109/TMC.2024.3373498).

[17] H. van Hasselt, A. Guez, and D. Silver, "Deep Reinforcement Learning with Double Q-learning", *Proc. of the AAAI Conference on Artificial Intelligence*, vol. 30, pp. 2094–2100, 2016 (https://doi.org/10.1609/AAAI.V30I1.10295).

[18] R.S. Sutton, D. Precup, and S. Singh, "Between MDPs and Semi-MDPs: A Framework for Temporal Abstraction in Reinforcement Learning", *Artificial Intelligence*, vol. 112, pp. 181–211, 1999 (https://doi.org/10.1016/S0004-3702(99)00052-1).

[19] F. Felten, E.-G. Talbi, and G. Danoy, "Multi-objective Reinforcement Learning Based on Decomposition: A Taxonomy and Framework", *Journal of Artificial Intelligence Research*, vol. 79, pp. 679–723, 2024 (https://doi.org/10.1613/JAIR.1.15702).

[20] R. Yang, X. Sun, and K. Narasimhan, "A Generalized Algorithm for Multi-objective Reinforcement Learning and Policy Adaptation", *arXiv*, 2019 (https://doi.org/10.48550/arXiv.1908.08342).

[21] C. Guo, G. Pleiss, Y. Sun, and K.Q. Weinberger, "On Calibration of Modern Neural Networks", *arXiv*, 2017 (https://doi.org/10.48550/arXiv.1706.04599).

Pham Nhat Quang, B.Eng.
https://orcid.org/0009-0002-1639-7615
E-mail: phnhatquang.2005@gmail.com
Vietnam National University, Hanoi, Vietnam
http://www.vnu.edu.vn

Nguyen Cong Binh, M.Sc.
https://orcid.org/0000-0002-2224-3164
E-mail: binhnc@eaut.edu.vn
East Asia University of Technology, Hanoi, Vietnam
https://eaut.edu.vn

Le Mai Nam, M.Sc.
https://orcid.org/0009-0007-3595-2377
E-mail: namlm@eaut.edu.vn
East Asia University of Technology, Hanoi, Vietnam
https://eaut.edu.vn

Nguyen Viet Hung, Ph.D.
https://orcid.org/0000-0002-9767-6749
E-mail: hungnv@eaut.edu.vn (corresponding author)
East Asia University of Technology, Hanoi, Vietnam
https://eaut.edu.vn