HETEROGENEITY-AWARE VALUE LEARNING FOR LONG-TERM RECOMMENDATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning (RL) provides a principled framework for optimizing long-term user engagement in recommender systems. Temporal-difference (TD) learning supports this objective by reusing trajectory segments through bootstrapped value estimation. In recommendation, sparse individual histories motivate knowledge transfer across users through a shared value function, yet heterogeneous interaction dynamics complicate this generalization. Updates induced by one user's experience may distort value estimates for others, with the resulting errors propagating through subsequent TD targets. To address this challenge, we propose Heterogeneity-Aware Value Learning (HAVL), which connects experience grouping with explicit control of shared updates. First, Heterogeneity-Aware Experience Grouping estimates expected reward, continuation probability, and successor-state features to organize experiences by their underlying interaction structure. Transfer-Constrained Value Learning then minimally adjusts proposed critic updates subject to reference TD-loss constraints defined by these groups. We characterize the grouping criterion through Bellman-backup discrepancies over a bounded affine continuation class and establish a local bound on group-level reference-loss increases. Experiments on real-world recommendation datasets demonstrate consistent cumulative-reward improvements across four RL backbones, complemented by an online evaluation in a production recommendation system.

1 INTRODUCTION

Long-term user engagement and retention are central objectives of recommender systems, yet they are not fully captured by immediate-feedback prediction (Zou et al., 2019; Liu et al., 2024). Reinforcement learning (RL) supports these objectives through cumulative-return optimization and has been studied extensively for long-term recommendation (Zheng et al., 2018; Xue et al., 2023; Cai et al., 2023; Xue et al., 2025; Wang et al., 2025). The cost of online interactions makes effective reuse of collected experience particularly important (Xin et al., 2020; Xue et al., 2023). Many of these approaches use temporal-difference (TD) learning to guide policy optimization with return estimates derived from observed feedback and learned successor values, without requiring complete rollouts (Ie et al., 2019; Xin et al., 2020).

TD learning supports policy improvement through trajectory stitching, whereby decisions learned from distinct trajectory segments can be combined into policies without complete demonstrations of the resulting behavior (Kostrikov et al., 2022). Bootstrapping provides a mechanism for this experience reuse: each training target combines an observed reward with a learned estimate of subsequent return. Through recursive Bellman backups, the value of a decision can therefore reflect continuations learned from other trajectories, rather than only the continuation observed in its original episode. In large state spaces, where trajectories rarely share identical states, this process additionally relies on generalization through a learned value function (Bortkiewicz et al., 2025).

In recommender systems, however, user heterogeneity complicates this generalization. Users with similar immediate responses may exhibit different patterns of preference evolution, continued interaction, and return visits (Cai et al., 2023; Xue et al., 2025). Together with large item catalogs, these heterogeneous dynamics create diverse user-specific interaction contexts that individual histories only sparsely cover (Zhang et al., 2024). Learning a shared value function across users is therefore an important means of improving statistical efficiency. Yet shared function approximation

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

can introduce update interference and representation aliasing in TD learning (Bengio et al., 2020; Kumar et al., 2022). In recommendation, a shared update may thus improve predictions for one user while degrading those for another with different long-term interaction dynamics. Since learned successor values also determine subsequent TD targets, the resulting estimation errors can propagate through recursive updates, distort the relative values of candidate items, and induce suboptimal recommendation policies (Kumar et al., 2020). The challenge is thus to preserve the benefits of cross-user sharing while limiting harmful transfer and its propagation through bootstrapped value learning.

To make shared learning more responsive to user heterogeneity, grouping-based approaches organize interaction contexts and regularize their representations, as exemplified by AURO's value-based state abstraction (Xue et al., 2025). However, similarity in current value does not imply similarity in the reward and transition dynamics underlying that value. Contexts assigned to the same group may therefore induce different TD targets as successor-value estimates evolve. Optimization-based approaches address interference by constraining or modifying gradients across learning objectives, as in GEM, PCGrad, and CAGrad (Lopez-Paz & Ranzato, 2017; Yu et al., 2020; Liu et al., 2021). These update rules regulate transfer between specified objectives, but do not themselves determine which recommendation experiences should define those objectives. **A remaining question is how to identify experiences with compatible long-term interaction structure and use this information to constrain cross-user transfer during value learning.**

To address this question, we propose Heterogeneity-Aware Value Learning (HAVL), comprising two complementary modules. First, *Heterogeneity-Aware Experience Grouping* estimates expected reward, continuation probability, and successor-state features for each interaction context. These estimates characterize the reward and transition structure underlying long-term value and provide a criterion for grouping experiences. To regulate the influence of shared updates on the resulting groups, we use *Transfer-Constrained Value Learning*, which minimally adjusts proposed TD updates subject to group-level reference-loss constraints. Our contributions are threefold:

- We investigate how user heterogeneity complicates cross-user sharing in long-term recommendation, and formulate the joint problem of identifying compatible experience and constraining its effects on shared value learning.
- We propose an experience-grouping method that organizes interactions by estimated reward and transition characteristics rather than current value similarity. We provide a Bellman-backup interpretation of the resulting grouping criterion.
- We develop a constrained value-learning method that translates experience groups into explicit constraints on shared TD updates. It combines minimal update correction with finite-step verification to limit increases in group-level reference TD losses.

2 Related Work

Reinforcement learning for recommendation. RL-based recommendation optimizes cumulative engagement and retention beyond immediate feedback. Value-based methods learn long-term returns for news recommendation and sequential engagement (Zheng et al., 2018; Zou et al., 2019), while SlateQ decomposes slate values to make TD learning tractable (Ie et al., 2019). Actor–critic approaches similarly rely on bootstrapped critics for policy improvement (Xue et al., 2023; Cai et al., 2023). Extensions integrate TD objectives with self-supervised recommendation (Xin et al., 2020) or conservative value estimation with evidential exploration (Wang et al., 2025). TD value learning is thus a recurring component across different recommendation formulations; our focus is how its updates generalize across heterogeneous users.

Stitching and interference in TD learning. TD-based dynamic programming supports trajectory stitching by combining decisions learned from distinct experience segments (Kostrikov et al., 2022). Under function approximation, stitching also involves generalization and is not exclusive to TD methods (Bortkiewicz et al., 2025). With shared parameters, however, bootstrapped updates affect predictions beyond the sampled transition; studies relate these interactions to generalization and control instability (Bengio et al., 2020; Liu et al., 2023). Existing remedies target complementary aspects: DisCor reweights samples according to estimated target accuracy (Kumar et al., 2020),

DR3 counteracts feature aliasing across Bellman backups (Kumar et al., 2022), and gradient-control methods regulate conflicting task updates (Lopez-Paz & Ranzato, 2017; Yu et al., 2020; Liu et al., 2021). MTRL-CG additionally groups tasks by update-related affinity and trains group-specific policies and critics (Meng et al., 2026). These studies motivate selective control of shared updates, rather than indiscriminate suppression of cross-sample influence.

User heterogeneity in RL-based recommendation. RL recommenders account for user differences through differentiated policies, exploration, and state representations. RLUR learns separate policies for high- and low-activity users (Cai et al., 2023), while UOEP uses return-distribution quantiles to guide group-oriented exploration (Zhang et al., 2024). AURO addresses behavioral non-stationarity through value-based state abstraction and guarded exploration, attracting states with similar estimated values and separating others (Xue et al., 2025). However, activity or return similarity alone does not ensure compatible Bellman backups: equal current values can conceal different rewards and successor dynamics. Geometric attraction and separation also do not directly constrain the TD-loss changes caused by a shared update. Bisimulation and behavioral similarity representations provide reward-, transition-, or policy-based notions of relatedness (Zhang et al., 2021; Agarwal et al., 2021); value equivalence compares Bellman updates relative to specified function and policy classes (Grimm et al., 2020). Our work, HAVL, adopts this structural perspective to group action-conditioned experiences by expected reward, continuation probability, and successor features, rather than current value alone. Group-level reference-loss constraints then regulate shared TD updates, providing explicit control over their effects beyond representation geometry.

3 PRELIMINARIES

We first formalize the recommendation objective, then describe the shared TD learning process whose cross-user influence we aim to control.

Long-term recommendation. We model recommendation as a discounted Markov decision process. The state $s_t = (u, h_t, c_t)$ contains user information u, interaction history h_t, and available context c_t. An action $a_t \in \mathcal{A}(s_t)$ selects an item from a candidate set. The environment produces reward r_t, successor state s_{t+1}, and termination indicator d_t. The history representation is assumed sufficient for the Markov model; omitted user factors can violate this assumption.

For policy π and discount $\gamma \in [0, 1)$, with zero rewards after termination,

$$Q^\pi(s,a) = \mathbb{E}_\pi\left[\sum_{\ell=0}^{\infty} \gamma^\ell r_{t+\ell} \mid s_t = s, a_t = a\right], \qquad J(\pi) = \mathbb{E}_{s_0\sim\mu_0, a_0\sim\pi} Q^\pi(s_0, a_0). \tag{1}$$

The objective is to maximize J under the evaluation distribution μ_0. User heterogeneity appears in the conditional rewards and transitions, even when different users encounter the same item. A true terminal transition has $d = 1$; a logging cutoff does not automatically terminate a continuing objective.

Shared value learning and Bellman backups. Rather than fit a separate value function for each user, we train one user-conditioned critic Q_θ on pooled transitions. Let $x = (s, a)$ denote a state–action pair. The action-conditioned Bellman backup of a continuation-value function $V : \mathcal{S} \to \mathbb{R}$ is

$$(\mathcal{T}V)(s,a) = \bar{r}(s,a) + \gamma\mathbb{E}[(1-d)V(s') \mid s, a]. \tag{2}$$

Here $\bar{r}(s,a) = \mathbb{E}[r \mid s, a]$, and $\mathcal{T}$ maps a state-value function to state–action backup values. For a fixed policy, $V^\pi_Q(s) = \mathbb{E}_{a\sim\pi(\cdot|s)}Q(s,a)$ gives the policy operator $\mathcal{T}^\pi Q = \mathcal{T}V^\pi_Q$, with $Q^\pi = \mathcal{T}^\pi Q^\pi$. For Q-learning, $V^*_Q(s) = \max_{a\in\mathcal{A}(s)} Q(s,a)$ gives the optimality operator $\mathcal{T}^* Q = \mathcal{T}V^*_Q$. Policy dependence is represented through V, while $\mathcal{T}$ is determined by the reward and transition dynamics.

For Q-learning, a target network supplies $V_{\bar{\theta}} = V^*_{Q_{\bar{\theta}}}$. For a mixed-user minibatch $\mathcal{D}_t$, ordinary TD learning uses

$$y_i = r_i + \gamma(1 - d_i)V_{\bar{\theta}}(s'_i), \qquad \delta_i = y_i - Q_\theta(x_i),$$
$$L_{\mathrm{TD}}(\theta) = \frac{1}{|\mathcal{D}_t|} \sum_{i \in \mathcal{D}_t} \frac{\delta_i^2}{2}. \tag{3}$$

Targets are fixed during each optimizer step. Every sample retains its own reward and successor, but its update can change other users' predictions through the shared parameters. The expected target at context x is $(\mathcal{T}V_{\bar{\theta}})(x)$: compatibility between two contexts therefore depends on how their rewards and transitions respond to the continuation function, not only on their current Q_θ values. We use this observation to construct a sharing criterion in Section 4.1, followed by a test of the actual update in Section 4.2.

4 Heterogeneity-Aware Value Learning

Our framework, HAVL, adapts cross-user sharing to the interaction structure underlying long-term value. As illustrated in Figure 1, it comprises two complementary modules. First, *Heterogeneity-Aware Experience Grouping* estimates the reward and continuation characteristics of state–action contexts to group experiences with similar Bellman-backup structure. *Transfer-Constrained Value Learning* then uses the resulting groups to define reference TD objectives and minimally adjusts proposed critic updates subject to group-level loss tolerances.

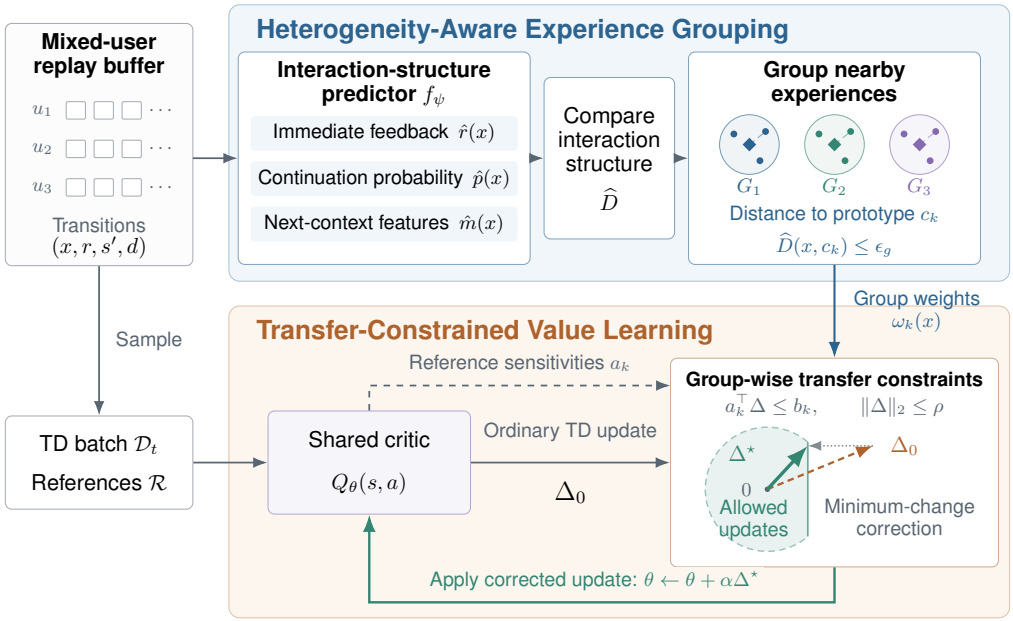


Figure 1: Overview of HAVL. Experiences are grouped according to their predicted reward and transition characteristics. The resulting groups define transfer constraints that yield a minimum-change correction from the ordinary TD proposal Δ_0 to $\Delta^\star$. The accepted step $\alpha\Delta^\star$ is then applied to the shared critic.

4.1 Heterogeneity-Aware Experience Grouping

The expected TD target depends on immediate reward and the distribution of subsequent interaction. We summarize these quantities with a descriptor $b(x) = (\bar{r}(x), \gamma p(x), m(x))$. For a fixed, bounded state-feature map $z : \mathcal{S} \to \mathbb{R}^q$, we define

$$\bar{r}(x) = \mathbb{E}[r \mid x],$$
$$p(x) = \mathbb{E}[1 - d \mid x], \tag{4}$$
$$m(x) = \mathbb{E}[\gamma(1 - d)z(s') \mid x].$$

Here, $\bar{r}(x)$ is the expected immediate reward, $p(x)$ is the probability of continued interaction, and $m(x)$ summarizes successor-state features weighted by continuation and discounting. $z(s') \in \mathbb{R}^q$ denotes the feature representation of the next state s', obtained using the fixed feature map z. To estimate these interaction characteristics from observed transitions, we train a three-head auxiliary predictor with outputs $(\hat{r}_\psi, \hat{p}_\psi, \hat{m}_\psi)$ by minimizing

$$L_{\text{aux}} = \mathbb{E}_{(x,r,s',d)}\big[\lambda_r(\hat{r}-r)^2 + \lambda_p \,\text{BCE}(\hat{p}, 1-d) + \lambda_m \|\hat{m} - \gamma(1-d)z(s')\|_2^2\big]. \tag{5}$$

The positive coefficients balance the prediction tasks. We construct z from pretrained state features and a fixed projection; the predictor and critic have separate parameters. Their training schedules are detailed in Appendix C.3.2.

To group experiences with similar reward and transition structure, we compare their predicted descriptors using

$$\widehat{D}_{ij} = |\hat{r}_i - \hat{r}_j| + C_0\gamma|\hat{p}_i - \hat{p}_j| + C_z\|\hat{m}_i - \hat{m}_j\|_2, \tag{6}$$

where $C_0, C_z \geq 0$ control sensitivity to continuation and successor features. Let $\{c_k\}_{k=1}^K$ denote descriptor prototypes initialized from a training reference pool, with $\widehat{D}(x, c_k)$ computed using the same weighted distance in descriptor space. Given a grouping radius ϵ_g and temperature $T_g > 0$, we define the soft membership of context x in group k as

$$\begin{aligned} \widetilde{\omega}_k(x) &= \mathbf{1}\{\widehat{D}(x, c_k) \leq \epsilon_g\} \exp[-\widehat{D}(x, c_k)/T_g], \\ \omega_k(x) &= \frac{\widetilde{\omega}_k(x)}{\sum_\ell \widetilde{\omega}_\ell(x)}. \end{aligned} \tag{7}$$

The radius ϵ_g restricts membership to nearby prototypes, while T_g controls the concentration of the membership weights. Assignments are defined at the state–action level, allowing different contexts of the same user to receive different memberships. Contexts outside all prototype neighborhoods are assigned zero membership weights but remain in the TD training objective.

Bellman interpretation. To relate the grouping criterion to value learning, consider the bounded affine continuation class and its worst-case backup discrepancy:

$$\begin{aligned} \mathcal{F} &= \{V_{c,w}(s) = c + w^\top z(s) : |c| \leq C_0,\ \|w\|_2 \leq C_z\}, \\ D_{\mathcal{F}}(x_i, x_j) &= \sup_{V \in \mathcal{F}} |(\mathcal{T}V)(x_i) - (\mathcal{T}V)(x_j)|. \end{aligned} \tag{8}$$

For any $V_{c,w} \in \mathcal{F}$, the Bellman backup is $(\mathcal{T}V_{c,w})(x) = \bar{r}(x) + \gamma p(x)c + m(x)^\top w$, yielding the following characterization.

Theorem 1 (Compatibility as backup discrepancy)**.** *For the class in Eq.* (8)*,*

$$D_{\mathcal{F}}(x_i, x_j) = |\bar{r}_i - \bar{r}_j| + C_0\gamma|p_i - p_j| + C_z\|m_i - m_j\|_2. \tag{9}$$

Define the weighted estimation error

$$e_i = |\hat{r}_i - \bar{r}_i| + C_0\gamma|\hat{p}_i - p_i| + C_z\|\hat{m}_i - m_i\|_2.$$

Then $|D_{\mathcal{F}}(x_i, x_j) - \widehat{D}_{ij}| \leq e_i + e_j$.

Remark (Grouping by target compatibility)**.** Two contexts assigned to the same prototype have expected TD targets within $2\epsilon_g + e_i + e_j$ for every $V \in \mathcal{F}$. The radius therefore controls target disagreement as continuation values vary within this class. Appendix B.1 gives the complete proof, the radius-based group bound, and a policy-value extension under an approximation assumption on the continuation class.

4.2 Transfer-Constrained Value Learning

Building on the experience groups constructed in Section 4.1, we regulate cross-user transfer by constraining the effect of shared critic updates on these groups. Each group defines a reference TD loss, whose gradient characterizes its local response to a proposed update. We then compute a minimum-change correction subject to group-level bounds on the predicted loss increase.

To construct these objectives, we sample a reference set $\mathcal{R}$ from training data independently of the TD batch, balancing user coverage. With membership weights and TD labels fixed, let $\ell_j(\theta) = \frac{1}{2}(y_j - Q_\theta(x_j))^2$ and define

$$L_k(\theta) = \frac{\sum_{j\in\mathcal{R}} \omega_k(x_j)\ell_j(\theta)}{\sum_{j\in\mathcal{R}} \omega_k(x_j)}, \qquad a_k = \nabla_\theta L_k(\theta). \tag{10}$$

Let $\mathcal{K}_t$ contain groups satisfying the reference-support threshold in Appendix C.3.3. For the proposed critic update Δ_0, $a_k^\top \Delta_0$ approximates the change in group k's reference loss. This update includes the learning rate and optimizer preconditioning.

We use these first-order loss estimates to constrain the update through a minimum-change projection (Lopez-Paz & Ranzato, 2017), with group-level budgets and a trust radius:

$$\begin{aligned} \Delta^\star = \arg\min_{\Delta} \quad & \tfrac{1}{2}\|\Delta - \Delta_0\|_2^2 \\ \text{subject to} \quad & a_k^\top \Delta \le b_k, \qquad k \in \mathcal{K}_t, \\ & \|\Delta\|_2 \le \rho. \end{aligned} \tag{11}$$

Here $b_k \ge 0$ bounds the first-order loss increase, and $\rho \ge 0$ limits the step norm. Feasible proposals are preserved. To enforce tolerances $\tau_k \ge 0$ on the nonlinear critic, we select a step size by backtracking until

$$L_k(\theta + \alpha\Delta^\star) \le L_k(\theta) + \tau_k, \qquad k \in \mathcal{K}_t, \tag{12}$$

with $\alpha \in \{1, \beta, \beta^2, \ldots\}$ and $0 < \beta < 1$. We then apply $\theta \leftarrow \theta + \alpha\Delta^\star$. The reference objectives remain fixed throughout the check; if no tested step is accepted, the update is skipped.

Theorem 2 (Local transfer bound)**.** *Suppose each fixed-target L_k has an H_k-Lipschitz gradient on the trust ball. Every feasible update in Eq. (11) satisfies*

$$L_k(\theta + \Delta) - L_k(\theta) \le b_k + \tfrac{1}{2}H_k\rho^2. \tag{13}$$

Remark (Controlling transfer at the update level)**.** Theorem 2 provides a local justification for transfer-constrained value learning: jointly constraining the first-order effect and magnitude of a shared critic update bounds the increase in each group's fixed-target reference TD loss. The budget b_k controls the predicted loss increase, while the radius ρ limits the deviation from this first-order approximation. The acceptance check enforces the prescribed loss tolerances directly, without estimating H_k. Appendix B.2 provides the proof and a population reference-loss extension; implementation details are given in Appendix C.3.

4.3 Overall Training and Inference

Algorithm 1 Heterogeneity-Aware Value Learning (HAVL)

Require: Critic Q_θ, target critic, auxiliary predictor and teacher, fixed z, training replay
1: Warm up the predictor using Eq. (5); initialize prototypes
2: **for** each training iteration **do**
3: Sample a mixed-user TD batch $\mathcal{D}_t$ and independent references $\mathcal{R}$
4: Construct fixed TD targets using Eq. (3) and group weights using Eq. (7)
5: Compute group losses and gradients a_k using Eq. (10)
6: Obtain the optimizer's proposed update Δ_0 from L_{TD}
7: Solve Eq. (11); select α using Eq. (12), or set $\alpha = 0$
8: Apply $\theta \leftarrow \theta + \alpha\Delta^\star$
9: Train the auxiliary predictor; update the target critic, teacher, and prototypes on schedule
10: **end for**
11: **return** Shared critic Q_θ for recommendation

Algorithm 1 summarizes the Q-learning instantiation. Grouping descriptors are supplied by an auxiliary teacher, a slowly updated copy of the predictor. The teacher, prototypes, and TD targets are refreshed between updates and held fixed throughout each critic update and its acceptance check, keeping the reference objectives unchanged. Each iteration requires one proposal gradient and one reference gradient per active group. At inference, the backbone's recommendation rule is unchanged; the auxiliary predictor and update controller are used only during training.

Table 1: Average cumulative reward (↑), reported as mean ± standard deviation over three independent training seeds. Best means are in bold.

	KuaiRand-1K				KuaiRand-27K			
Method	DQN	A2C	DDPG	TD3	DQN	A2C	DDPG	TD3
Base	8.5578 ±0.0486	8.7036 ±0.0417	8.6833 ±0.0280	9.0190 ±0.0385	9.1537 ±0.0468	9.4617 ±0.0415	9.2714 ±0.0289	9.8573 ±0.0217
User-uniform sampling	8.6413 ±0.0472	8.7789 ±0.0436	8.7247 ±0.0368	9.0762 ±0.0374	9.1968 ±0.0473	9.5036 ±0.0398	9.2478 ±0.0376	9.9134 ±0.0281
RLUR	8.7736 ±0.0465	8.9368 ±0.0389	8.9134 ±0.0412	9.2287 ±0.0361	9.3942 ±0.0417	9.7264 ±0.0432	9.5537 ±0.0384	10.1036 ±0.0279
AURO	8.8624 ±0.0413	9.0327 ±0.0478	8.9765 ±0.0387	9.3138 ±0.0326	9.5476 ±0.0471	9.8638 ±0.0368	9.6342 ±0.0463	10.1867 ±0.0314
GEM	8.8379 ±0.0524	8.9143 ±0.0481	9.0632 ±0.0426	9.2864 ±0.0389	9.4831 ±0.0516	9.6769 ±0.0474	9.7036 ±0.0419	10.1432 ±0.0378
DisCor	8.9068 ±0.0379	8.9726 ±0.0414	9.1087 ±0.0316	9.3641 ±0.0298	9.5634 ±0.0387	9.7741 ±0.0392	9.7568 ±0.0307	10.2274 ±0.0263
HAVL–Random	8.5864 ±0.0573	8.7238 ±0.0496	8.6741 ±0.0437	9.0376 ±0.0418	9.1642 ±0.0564	9.4367 ±0.0489	9.3038 ±0.0416	9.8469 ±0.0327
HAVL–Activity	8.9327 ±0.0396	9.0564 ±0.0423	9.0348 ±0.0385	9.3279 ±0.0367	9.6478 ±0.0376	9.8932 ±0.0394	9.7164 ±0.0381	10.2137 ±0.0296
HAVL–Value	8.8836 ±0.0487	9.0132 ±0.0469	9.1437 ±0.0374	9.3968 ±0.0412	9.6031 ±0.0462	9.8368 ±0.0473	9.8146 ±0.0369	10.2764 ±0.0387
HAVL (ours)	**9.0243** ±0.0382	**9.1276** ±0.0375	**9.2341** ±0.0358	**9.4726** ±0.0284	**9.7136** ±0.0451	**9.9647** ±0.0362	**9.9378** ±0.0374	**10.3589** ±0.0286

5 EXPERIMENTS

Our evaluation examines recommendation performance, the contributions of grouping and update control, and cross-user sharing.

5.1 EXPERIMENTAL SETUP

Datasets and environments. We consider two recommendation settings based on KuaiRand-1K and KuaiRand-27K, which provide sequential user–video interactions and rich user and item features (Gao et al., 2022). KuaiRand-1K contains a subset of users sampled from KuaiRand-27K, allowing evaluation at different population and data scales within the same recommendation domain. We adopt KuaiSim for interactive policy evaluation (Zhao et al., 2023). At each step, the agent recommends one item and receives simulated click feedback; a feedback-dependent termination rule determines the session duration. The response model and termination configuration remain fixed across compared methods within each setting.

Backbones and baselines. We evaluate HAVL with four RL backbones: DQN (Mnih et al., 2015), A2C (Mnih et al., 2016), DDPG (Lillicrap et al., 2016), and TD3 (Fujimoto et al., 2018). A2C denotes the synchronous variant of advantage actor–critic. Comparisons include the unmodified backbone and user-uniform sampling, which balances user contributions during training. We further consider adaptations of RLUR's activity-specific policy learning (Cai et al., 2023) and AURO's value-based representation regularization (Xue et al., 2025), together with GEM's reference-gradient constraints (Lopez-Paz & Ranzato, 2017) and DisCor's error-based distribution correction (Kumar et al., 2020). These comparisons examine user heterogeneity, shared-update interference, and corrective feedback through backbone-specific adaptations, rather than reproductions of the complete original systems.

Grouping variants. To isolate the contribution of heterogeneity-aware experience grouping, we compare HAVL with three variants: random grouping, activity-based grouping, and value-based grouping. Activity is measured by users' interaction counts in the training data, whereas value-based grouping uses scalar critic predictions. All variants retain the same transfer-constrained update procedure, number of groups, and reference-sampling protocol, replacing only the grouping criterion.

Evaluation protocol. The primary metric is average cumulative reward, computed as

$$\overline{R} = \frac{1}{N_{\text{eval}}} \sum_{e=1}^{N_{\text{eval}}} \sum_{t=0}^{T_e-1} r_{e,t}, \tag{14}$$

where N_{eval} is the number of evaluation sessions and T_e is the length of session e. Rewards are accumulated without discounting for evaluation, with equal numbers of sessions assigned to each test user. Within each dataset–backbone pair, methods share the candidate-item support and environment-interaction budget. Appendix C.3 details the HAVL–DQN configuration and its fixed-budget checkpoint rule. Each configuration is evaluated across three independent training seeds, and results are summarized by the mean and standard deviation of the seed-level average rewards.

5.2 Recommendation Performance Comparison

Table 1 shows that HAVL achieves the highest average cumulative reward in all eight dataset–backbone settings, improving over Base by 4.87–7.19% and the strongest non-HAVL comparator in each column by 1.02–1.86%. User-uniform sampling yields limited gains, including a decrease for DDPG on KuaiRand-27K. Under the same update controller, HAVL outperforms activity- and value-based grouping, whereas random grouping remains close to Base. This pattern supports the importance of organizing experience by interaction structure for shared value learning.

5.3 Contributions of Grouping and Update Control

The grouping variants in Table 1 assess the choice of sharing criterion. Table 2 reports further component ablations. *Single-group control* pools all reference experiences into one group while retaining the update controller. We additionally ablate the reward, explicit continuation, and successor-feature distance terms individually, without changing the auxiliary predictor or update controller.

Removing update control recovers the Base critic update. Backbone, training budget, reference pool, and seed settings remain fixed within each comparison.

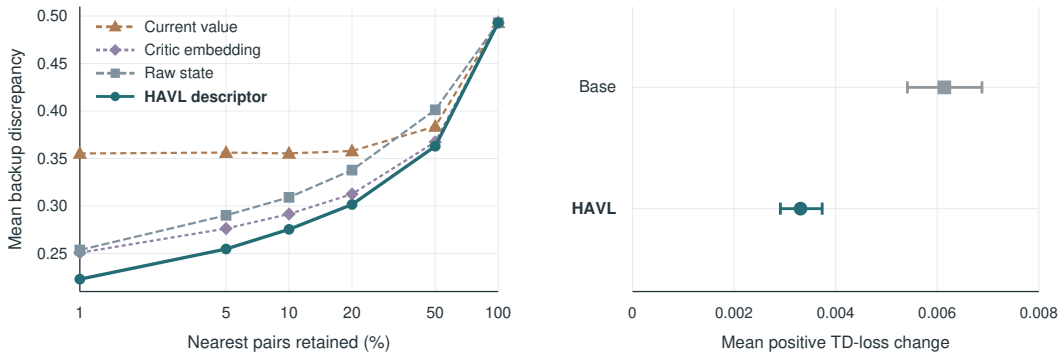
Table 2: Component ablations measured by average cumulative reward (↑). Values are mean ± standard deviation over three training seeds.

	KuaiRand-1K				KuaiRand-27K			
Variant	DQN	A2C	DDPG	TD3	DQN	A2C	DDPG	TD3
HAVL (full)	9.0243 ±0.0382	9.1276 ±0.0375	9.2341 ±0.0358	9.4726 ±0.0284	9.7136 ±0.0451	9.9647 ±0.0362	9.9378 ±0.0374	10.3589 ±0.0286
Single-group control	8.7351 ±0.0509	8.8605 ±0.0458	8.9036 ±0.0409	9.1959 ±0.0376	9.3553 ±0.0517	9.6528 ±0.0445	9.5313 ±0.0412	10.0429 ±0.0328
w/o reward term	8.9263 ±0.0444	9.0343 ±0.0421	9.1239 ±0.0393	9.3683 ±0.0341	9.5904 ±0.0482	9.8591 ±0.0410	9.7979 ±0.0399	10.2586 ±0.0317
w/o continuation term	8.9777 ±0.0401	9.0810 ±0.0390	9.1845 ±0.0371	9.4272 ±0.0306	9.6520 ±0.0460	9.9144 ±0.0381	9.8778 ±0.0384	10.3037 ±0.0299
w/o successor term	8.8797 ±0.0472	8.9877 ±0.0438	9.0744 ±0.0402	9.3274 ±0.0356	9.5456 ±0.0499	9.8088 ±0.0424	9.7512 ±0.0410	10.1984 ±0.0325
w/o update control (Base)	8.5578 ±0.0486	8.7036 ±0.0417	8.6833 ±0.0280	9.0190 ±0.0385	9.1537 ±0.0468	9.4617 ±0.0415	9.2714 ±0.0289	9.8573 ±0.0217

Full HAVL attains the highest mean reward in all eight settings. Single-group control yields 2.92–4.09% lower reward than full HAVL, yet remains 1.80–2.80% above Base, indicating that control helps without structured groups and learned grouping adds further gains. Every distance-term ablation also reduces reward relative to full HAVL: by 0.48–0.63% without explicit continuation, 0.97–1.41% without reward, and 1.53–1.88% without successor features. These results support contributions from all three distance terms. Among these term ablations, removing the successor-feature term yields the largest observed reward decrease across both datasets and all backbones.

5.4 Understanding Cross-User Sharing

We examine the experience similarities underlying grouping and the effect of update control using two diagnostics on KuaiRand-27K. First, we rank 200,000 same-action, cross-user context pairs by



(a) Experience similarity

(b) Shared-update effects

Figure 2: Diagnostics of cross-user sharing. Lower is better in both panels. (a) Mean absolute backup discrepancy for pairs retained by each distance, using 4,096 users, 8,192 contexts, 32 simulator draws per context, and eight continuation probes. All rankings coincide at 100% coverage. (b) Mean positive fixed-target TD-loss change on independent receiver-user contexts. Intervals are 95% bootstrap intervals over 32 proposal batches, conditional on one DQN seed and checkpoint. Appendix C.5 details both protocols and the harm comparison.

scalar value, critic representation, state features, or learned descriptors. Repeated simulator draws and eight fixed continuation functions provide a separate estimate of their Bellman-backup discrepancy. Figure 2(a) shows that progressively restricting descriptor-based selection identifies pairs with closer backup responses, whereas scalar-value selection changes little over the nearest 1–20% of pairs. This supports examining interaction structure beyond a single value prediction; the diagnostic evaluates pair selection rather than the complete grouping procedure.

Second, we evaluate ordinary and controlled updates from identical DQN learner states on audit contexts excluded from training, constraint construction, and acceptance checks. We measure cross-user harm as the positive part of each context's fixed-target TD-loss change, averaged within users and then equally across users. In Figure 2(b), HAVL reduces mean harm by 46.1% relative to Base.

5.5 Online Experiment

We evaluate HAVL for sequential rewarded-ad delivery in Tencent's WeChat Mini Programs, where the policy selects ad duration and content-unlocking rewards for short-drama sessions. In a seven-day online A/B test, HAVL improves click-through rate (CTR) by 1.49% and gross merchandise volume (GMV) by 2.29% relative to the production DQN baseline. Appendix C.4 details the application and evaluation design.

6 Conclusion

Effective cross-user sharing in long-term recommendation requires attention to both interaction structure and shared-update effects. We proposed HAVL, which groups experiences by reward and transition characteristics and minimally corrects TD updates under group-level reference-loss constraints. Our analysis characterizes the grouping criterion through Bellman-backup discrepancies and bounds local increases in reference TD losses. Experiments on two KuaiRand datasets across four RL backbones, together with an online evaluation, support the effectiveness of this approach.

AI USE STATEMENT

We used generative AI tools, including OpenAI Codex, to assist with hypothesis refinement, conceptual and methodological development, mathematical formulations and proof drafting, experimental design, method implementation and debugging, and interpretation of experimental results. These tools also supported literature discovery and synthesis, scientific figure preparation, and manuscript drafting and editing. The authors checked the cited sources and reviewed the mathematical assumptions and derivations. We checked algorithm descriptions against the available implementation and numerical statements against the reported result summaries. The authors remain responsible for the accuracy, originality, and integrity of the final manuscript and accompanying artifacts.

REFERENCES

Rishabh Agarwal, Marlos C. Machado, Pablo Samuel Castro, and Marc G. Bellemare. Contrastive behavioral similarity embeddings for generalization in reinforcement learning. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2101.05265.

Emmanuel Bengio, Joelle Pineau, and Doina Precup. Interference and generalization in temporal difference learning. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pp. 767–777. PMLR, 2020. URL https://proceedings.mlr.press/v119/bengio20a.html.

Michał Bortkiewicz, Władysław Pałucki, Mateusz Ostaszewski, and Benjamin Eysenbach. Is temporal difference learning the gold standard for stitching in RL?, 2025. URL https://arxiv.org/abs/2510.21995v1. arXiv:2510.21995v1.

Qingpeng Cai, Shuchang Liu, Xueliang Wang, Tianyou Zuo, Wentao Xie, Bin Yang, Dong Zheng, Peng Jiang, and Kun Gai. Reinforcing user retention in a billion scale short video recommender system. In *The Web Conference, Industry Track*, 2023. URL https://arxiv.org/abs/2302.01724.

Scott Fujimoto, Herke van Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1587–1596. PMLR, 2018. URL https://proceedings.mlr.press/v80/fujimoto18a.html.

Chongming Gao, Shijun Li, Yuan Zhang, Jiawei Chen, Biao Li, Wenqiang Lei, Peng Jiang, and Xiangnan He. KuaiRand: An unbiased sequential recommendation dataset with randomly exposed videos. In *Proceedings of the ACM International Conference on Information and Knowledge Management*, pp. 3953–3957, 2022. doi: 10.1145/3511808.3557624. URL https://arxiv.org/abs/2208.08696.

Christopher Grimm, André Barreto, Satinder Singh, and David Silver. The value equivalence principle for model-based reinforcement learning. In *Advances in Neural Information Processing Systems*, 2020. URL https://arxiv.org/abs/2011.03506.

Eugene Ie, Vihan Jain, Jing Wang, Sanmit Narvekar, Ritesh Agarwal, Rui Wu, Heng-Tze Cheng, Tushar Chandra, and Craig Boutilier. SlateQ: A tractable decomposition for reinforcement learning with recommendation sets. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, pp. 2592–2599, 2019. doi: 10.24963/ijcai.2019/360. URL https://www.ijcai.org/proceedings/2019/360.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit Q-learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=EblVBDNalKu.

Aviral Kumar, Abhishek Gupta, and Sergey Levine. DisCor: Corrective feedback in reinforcement learning via distribution correction. In *Advances in Neural Information Processing Systems*, volume 33, pp. 18560–18572, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/d7f426ccbc6db7e235c57958c21c5dfa-Abstract.html.

Aviral Kumar, Rishabh Agarwal, Tengyu Ma, Aaron Courville, George Tucker, and Sergey Levine. DR3: Value-based deep reinforcement learning requires explicit regularization. In *International Conference on Learning Representations*, 2022. URL https://iclr.cc/virtual/2022/poster/6927.

Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. In *International Conference on Learning Representations*, 2016. URL https://arxiv.org/abs/1509.02971.

Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. Conflict-averse gradient descent for multi-task learning. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/9d27fdf2477ffbff837d73ef7ae23db9-Abstract.html.

Vincent Liu, Han Wang, Ruo Yu Tao, Khurram Javed, Adam White, and Martha White. Measuring and mitigating interference in reinforcement learning. In *Proceedings of the 2nd Conference on Lifelong Learning Agents*, volume 232, pp. 781–795. PMLR, 2023. URL https://proceedings.mlr.press/v232/liu23a.html.

Ziru Liu, Shuchang Liu, Zijian Zhang, Qingpeng Cai, Xiangyu Zhao, Kesen Zhao, Lantao Hu, Peng Jiang, and Kun Gai. Sequential recommendation for optimizing both immediate feedback and long-term retention. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1872–1882, 2024. doi: 10.1145/3626772.3657829. URL https://arxiv.org/abs/2404.03637.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*, 2017. URL https://arxiv.org/abs/1706.08840.

Wenjia Meng, Teng Zhang, Haoliang Sun, and Yilong Yin. MTRL-CG: Multi-task reinforcement learning method with spectral clustering-based task grouping. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(43):36723–36731, 2026. doi: 10.1609/aaai.v40i43.40997. URL https://ojs.aaai.org/index.php/AAAI/article/view/40997.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015. doi: 10.1038/nature14236. URL https://www.nature.com/articles/nature14236.

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *Proceedings of the 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 1928–1937. PMLR, 2016. URL https://proceedings.mlr.press/v48/mniha16.html.

Dingrong Wang, Krishna Prasad Neupane, Ervine Zheng, and Qi Yu. Looking into user's long-term interests through the lens of conservative evidential learning. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/eb1a323fa10d4102ff13422476a744ff-Abstract-Conference.html.

Xin Xin, Alexandros Karatzoglou, Ioannis Arapakis, and Joemon M. Jose. Self-supervised reinforcement learning for recommender systems. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 931–940, 2020. doi: 10.1145/3397271.3401147. URL https://arxiv.org/abs/2006.05779.

Wanqi Xue, Qingpeng Cai, Ruohan Zhan, Dong Zheng, Peng Jiang, Kun Gai, and Bo An. ResAct: Reinforcing long-term engagement in sequential recommendation with residual actor. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=HmPOzJQhbwg.

Zhenghai Xue, Qingpeng Cai, Bin Yang, Lantao Hu, Peng Jiang, Kun Gai, and Bo An. AURO: Reinforcement learning for adaptive user retention optimization in recommender systems. In *Proceedings of the ACM Web Conference*, pp. 391–401, 2025. doi: 10.1145/3696410.3714956. URL https://arxiv.org/abs/2310.03984.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 5824–5836, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/3fe78a8acf5fda99de95303940a2420c-Abstract.html.

Amy Zhang, Rowan McAllister, Roberto Calandra, Yarin Gal, and Sergey Levine. Learning invariant representations for reinforcement learning without reconstruction. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2006.10742.

Changshuo Zhang, Sirui Chen, Xiao Zhang, Sunhao Dai, Weijie Yu, and Jun Xu. Reinforcing long-term performance in recommender systems with user-oriented exploration policy. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1850–1860, 2024. doi: 10.1145/3626772.3657714. URL https://doi.org/10.1145/3626772.3657714.

Kesen Zhao, Shuchang Liu, Qingpeng Cai, Xiangyu Zhao, Ziru Liu, Dong Zheng, Peng Jiang, and Kun Gai. KuaiSim: A comprehensive simulator for recommender systems. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/8c7f8f98f9a8f5650922dd4545254f28-Abstract.html. Datasets and Benchmarks Track.

Guanjie Zheng, Fuzheng Zhang, Zihan Zheng, Yang Xiang, Nicholas Jing Yuan, Xing Xie, and Zhenhui Li. DRN: A deep reinforcement learning framework for news recommendation. In *Proceedings of the 2018 World Wide Web Conference*, pp. 167–176, 2018. doi: 10.1145/3178876.3185994. URL https://doi.org/10.1145/3178876.3185994.

Lixin Zou, Long Xia, Zhuoye Ding, Jiaxing Song, Weidong Liu, and Dawei Yin. Reinforcement learning to optimize long-term user engagement in recommender systems. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 2810–2818, 2019. doi: 10.1145/3292500.3330668. URL https://arxiv.org/abs/1902.05570.

Appendix

Table of Contents

A Summary of Notation 14

B Proofs of Theorems 15

B.1 Proof of Theorem 1 15

B.2 Proof of Theorem 2 16

B.3 Why Grouping and Update Control Are Complementary 17

C Detailed Experimental Setup 17

C.1 Offline Recommendation Settings 17

C.2 Training and Evaluation Protocol 17

C.3 Implementation Details 18

C.4 Online Experiment Scenario 21

C.5 Cross-User Sharing Diagnostics 21

D Source Code 22

A SUMMARY OF NOTATION

Table 3 summarizes the notation used in the formulation, method, and analysis.

Table 3: Summary of notation.

Notation	Description
$s_t = (u, h_t, c_t)$	State containing user information, interaction history, and current context.
$a \in \mathcal{A}(s),\ x = (s, a)$	Candidate action and state–action context.
$r,\ s',\ d$	Observed reward, successor state, and termination indicator.
$\gamma,\ \pi,\ \mu_0$	Discount factor, policy, and initial evaluation-state distribution.
$Q^\pi,\ V^\pi,\ J(\pi)$	Policy action value, state value, and expected discounted-return objective.
$Q_\theta,\ Q_{\bar{\theta}}$	Shared critic and target critic.
$\mathcal{T},\ V$	Action-conditioned Bellman backup and continuation-value function.
$\mathcal{D}_t,\ \mathcal{R}$	Mixed-user TD minibatch and independent training reference set.
$y_i,\ \delta_i,\ L_{\mathrm{TD}}$	Fixed TD target, residual, and batch TD objective.
$z(s) \in \mathbb{R}^q$	Fixed, bounded state-feature representation of dimension q.
$\bar{r}(x),\ p(x),\ m(x)$	Expected reward, continuation probability, and discounted continuation-weighted successor-feature mean.
$b(x)$	Interaction descriptor $(\bar{r}(x), \gamma p(x), m(x))$.
$\hat{r}_\psi,\ \hat{p}_\psi,\ \hat{m}_\psi$	Three outputs of the auxiliary predictor with parameters ψ.
$L_{\mathrm{aux}},\ \lambda_r, \lambda_p, \lambda_m$	Auxiliary prediction objective and positive task weights.
$\mathcal{F},\ C_0,\ C_z$	Bounded affine continuation class and its coefficient bounds.
$D_{\mathcal{F}},\ \widehat{D}_{ij},\ e_i$	True backup discrepancy, estimated distance, and weighted descriptor error.
$K,\ c_k,\ \epsilon_g,\ T_g$	Number of prototypes, descriptor prototype, grouping radius, and temperature.
$\omega_k(x),\ \mathcal{K}_t$	Soft group membership and active reference groups.
$L_k,\ a_k = \nabla_\theta L_k$	Group-level reference TD loss and its gradient.
$\Delta_0,\ \Delta^\star$	Ordinary optimizer update and minimum-change corrected update.
$b_k,\ \rho,\ \tau_k$	First-order loss budget, trust radius, and finite-step loss tolerance.
$\alpha,\ \beta$	Accepted step multiplier and backtracking shrinkage factor.
$H_k,\ \bar{H}_k,\ \zeta_k$	Reference and population smoothness constants, and reference-gradient error bound.
$\overline{R},\ N_{\mathrm{eval}},\ T_e$	Average cumulative reward, evaluation-session count, and length of session e.

B PROOFS OF THEOREMS

B.1 PROOF OF THEOREM 1

Throughout this subsection, z is fixed and bounded, rewards are integrable, and conditional transition distributions are fixed. The moments $\bar{r}_i, p_i, m_i$ are defined in Eq. (4). All contexts use the same continuation class

$$\mathcal{F} = \{V_{c,w}(s) = c + w^\top z(s) : |c| \le C_0,\ \|w\|_2 \le C_z\}, \qquad C_0, C_z \ge 0.$$

Theorem 1 (Compatibility as backup discrepancy, restated)**.** *For any two contexts x_i, x_j, the worst-case Bellman-backup discrepancy over $\mathcal{F}$ satisfies*

$$\begin{aligned} D_{\mathcal{F}}(x_i, x_j) &= \sup_{V \in \mathcal{F}} |(\mathcal{T}V)(x_i) - (\mathcal{T}V)(x_j)| \\ &= |\bar{r}_i - \bar{r}_j| + C_0\gamma|p_i - p_j| + C_z\|m_i - m_j\|_2. \end{aligned}$$

For predicted moments, define

$$e_i = |\hat{r}_i - \bar{r}_i| + C_0\gamma|\hat{p}_i - p_i| + C_z\|\hat{m}_i - m_i\|_2.$$

The estimated distance in Eq. (6) then obeys $|D_{\mathcal{F}}(x_i, x_j) - \widehat{D}_{ij}| \le e_i + e_j$.

Proof. First, define $\Delta r = \bar{r}_i - \bar{r}_j$, $\Delta p = p_i - p_j$, and $\Delta m = m_i - m_j$. Linearity of conditional expectation gives, for every $V_{c,w} \in \mathcal{F}$,

$$(\mathcal{T}V_{c,w})(x_i) - (\mathcal{T}V_{c,w})(x_j) = \Delta r + \gamma \Delta p\, c + \Delta m^\top w, \tag{15}$$

$$\left|\Delta r + \gamma \Delta p\, c + \Delta m^\top w\right| \le |\Delta r| + \gamma C_0 |\Delta p| + C_z \|\Delta m\|_2. \tag{16}$$

The second line uses the triangle inequality and Cauchy–Schwarz. For attainability, select $\sigma \in \{-1, 1\}$ with $\sigma \Delta r = |\Delta r|$; when $\Delta r = 0$, either sign suffices. Choose

$$c^\star = \sigma C_0 \operatorname{sign}(\Delta p), \qquad w^\star = \begin{cases} \sigma C_z \Delta m / \|\Delta m\|_2, & \Delta m \neq 0, \\ 0, & \Delta m = 0. \end{cases}$$

Take $\operatorname{sign}(0) = 0$. These choices are feasible and make all nonzero terms have sign σ, attaining the upper bound. This proves Eq. (9), including zero coefficients or differences.

For the estimation bound, apply the reverse triangle inequality to each distance component. In particular,

$$\left||\hat{r}_i - \hat{r}_j| - |\bar{r}_i - \bar{r}_j|\right| \le |\hat{r}_i - \bar{r}_i| + |\hat{r}_j - \bar{r}_j|,$$

and the same argument holds for the scalar survival component and the Euclidean successor component. Summing their weighted bounds yields $|\widehat{D}_{ij} - D_{\mathcal{F}}(x_i, x_j)| \le e_i + e_j$. □

Remark (Within-group backup compatibility)**.** Both $D_{\mathcal{F}}$ and $\widehat{D}$ are pseudometrics: nonnegativity, symmetry, and the triangle inequality follow from their weighted norm form, although distinct contexts can share the same descriptor. If $\omega_k(x_i) > 0$ and $\omega_k(x_j) > 0$, the radius rule gives

$$\widehat{D}_{ij} \le \widehat{D}(x_i, c_k) + \widehat{D}(c_k, x_j) \le 2\epsilon_g.$$

Consequently, $D_{\mathcal{F}}(x_i, x_j) \le 2\epsilon_g + e_i + e_j$. This is a support property of each group with the teacher and prototypes fixed, regardless of the nonzero membership weights.

Corollary 1 (Connection to policy values)**.** *If $\inf_{V \in \mathcal{F}} \|V^\pi - V\|_\infty \le \epsilon_V$, then*

$$|Q^\pi(x_i) - Q^\pi(x_j)| \le \widehat{D}_{ij} + e_i + e_j + \gamma(p_i + p_j)\epsilon_V. \tag{17}$$

Proof. For any $\varepsilon > 0$, the approximation assumption gives $V_0 \in \mathcal{F}$ satisfying $\|V^\pi - V_0\|_\infty \le \epsilon_V + \varepsilon$. Because rewards cancel,

$$\begin{aligned} |(\mathcal{T}V^\pi)(x_i) - (\mathcal{T}V_0)(x_i)| &\le \gamma \mathbb{E}[(1-d)|V^\pi(s') - V_0(s')| \mid x_i] \\ &\le \gamma p_i(\epsilon_V + \varepsilon). \end{aligned}$$

Insert $\mathcal{T}V_0$ at both contexts, apply the triangle inequality and Theorem 1, and use $Q^\pi = \mathcal{T}V^\pi$. The result is

$$|Q^\pi(x_i) - Q^\pi(x_j)| \le \widehat{D}_{ij} + e_i + e_j + \gamma(p_i + p_j)(\epsilon_V + \varepsilon).$$

Letting $\varepsilon \downarrow 0$ proves the claim without requiring attainment of the infimum. □

Remark (Scope of the continuation class)**.** The descriptor characterizes backups over $\mathcal{F}$, not the full successor distribution. A symmetric distribution on $z = -1, +1$ and a point mass at $z = 0$ have equal feature means but different expectations of $V(z) = z^2$. Corollary 1 therefore retains the policy-value approximation error ϵ_V. Likewise, the pointwise descriptor errors e_i are not supplied by an average auxiliary prediction loss.

B.2 Proof of Theorem 2

Theorem 2 (Local transfer bound, restated)**.** *Fix the reference samples, group memberships, and TD targets. Suppose each group loss L_k has an H_k-Lipschitz gradient on the trust ball $\{\theta+v : \|v\|_2 \le \rho\}$, and let $a_k = \nabla L_k(\theta)$. Every update satisfying*

$$a_k^\top \Delta \le b_k \quad (k \in \mathcal{K}_t), \qquad \|\Delta\|_2 \le \rho$$

obeys

$$L_k(\theta + \Delta) - L_k(\theta) \le b_k + \tfrac{1}{2} H_k \rho^2, \qquad k \in \mathcal{K}_t.$$

Proof. Because $\|\Delta\|_2 \le \rho$, the segment $\theta + t\Delta$, $0 \le t \le 1$, lies in the trust ball. The fundamental theorem of calculus gives

$$\begin{aligned} L_k(\theta + \Delta) - L_k(\theta) &= a_k^\top \Delta + \int_0^1 [\nabla L_k(\theta + t\Delta) - a_k]^\top \Delta \, dt && (18)\\ &\le a_k^\top \Delta + \int_0^1 H_k t \|\Delta\|^2 \, dt && (19)\\ &= a_k^\top \Delta + \tfrac{1}{2} H_k \|\Delta\|^2 && (20)\\ &\le b_k + \tfrac{1}{2} H_k \rho^2. && (21) \end{aligned}$$

The first inequality combines Cauchy–Schwarz with the Lipschitz-gradient assumption; the last applies the linear budget and trust-radius constraints. □

Corollary 2 (Population reference losses)**.** *Let $\mathcal{L}_k$ be the population loss under a fixed-target reference distribution. Suppose it has a $\bar{H}_k$-Lipschitz gradient on the trust ball and $\|\nabla \mathcal{L}_k(\theta) - a_k\|_2 \le \zeta_k$. Every update feasible in Eq. (11) satisfies*

$$\mathcal{L}_k(\theta + \Delta) - \mathcal{L}_k(\theta) \le b_k + \zeta_k \rho + \tfrac{1}{2} \bar{H}_k \rho^2.$$

Proof. Apply the smoothness argument above with $\bar{H}_k$ and use

$$\begin{aligned} \nabla \mathcal{L}_k(\theta)^\top \Delta &= a_k^\top \Delta + (\nabla \mathcal{L}_k(\theta) - a_k)^\top \Delta \\ &\le b_k + \zeta_k \rho. \end{aligned}$$

Adding the remainder $\bar{H}_k \|\Delta\|_2^2 / 2 \le \bar{H}_k \rho^2 / 2$ proves the result. The bound applies to the specified fixed-target reference distribution. □

Remark (Projection and finite-step acceptance)**.** For $b_k \ge 0$ and finite $\rho \ge 0$, the feasible set of Eq. (11) is nonempty, compact, and convex. Its strictly convex objective therefore has a unique minimizer, which equals Δ_0 whenever the proposal is feasible. Backtracking preserves feasibility because, for $\alpha \in [0, 1]$,

$$a_k^\top (\alpha \Delta) \le \alpha b_k \le b_k, \qquad \|\alpha \Delta\|_2 \le \rho.$$

Continuity ensures that sufficiently small steps satisfy Eq. (12) when every $\tau_k > 0$. A finite search or zero tolerances may instead require the zero-update fallback. For ReLU critics, the smoothness bound applies on trust balls contained within a fixed activation region; the direct acceptance check does not require this condition. These statements concern group-average reference losses with fixed targets; they do not ensure that every context improves or that policy return increases.

B.3 Why Grouping and Update Control Are Complementary

Equal scalar value, different backup response. Let two contexts have reward zero and survive with probability one. Their deterministic successors have scalar features $z(s'_i) = 1$ and $z(s'_j) = -1$. Under $V_0 \equiv 0$, both backups are zero. Under $V_1(s) = wz(s)$ for $0 < w \le C_z$, their backups differ by $2\gamma w$. The compatibility distance is $2\gamma C_z$. Thus equality for one continuation value is weaker than compatibility over a class, even without reward differences or estimation noise.

Identical descriptors, harmful current transfer. Consider two terminal contexts with deterministic reward zero. Their true descriptors and targets are identical: $D_{\mathcal{F}}(x_i, x_j) = 0$ and $y_i = y_j = 0$. Let the shared predictor be $Q_\theta(x_i) = \theta + 1$ and $Q_\theta(x_j) = \theta - 1$. At $\theta = 0$, a gradient step on i changes θ to $-\eta$. Its own squared loss decreases for $0 < \eta < 2$, but context j's loss increases by

$$\tfrac{1}{2}(-\eta - 1)^2 - \tfrac{1}{2}(-1)^2 = \eta + \tfrac{1}{2}\eta^2 > 0.$$

This example isolates the effect of current prediction errors; it does not claim every sufficiently expressive critic exhibits this conflict. It disproves the implication that zero compatibility distance alone certifies a beneficial update.

Derivation of local influence. With $\ell_j = \delta_j^2/2$ and frozen y_j, $\nabla \ell_j = -\delta_j \nabla Q_j$. A descent step from i is $\Delta_i = -\eta \nabla \ell_i = \eta \delta_i \nabla Q_i$. Taylor expansion yields

$$\begin{aligned} \ell_j(\theta + \Delta_i) - \ell_j(\theta) &= \nabla \ell_j^\top \Delta_i + O(\|\Delta_i\|^2) \\ &= -\eta \delta_i \delta_j \nabla Q_j^\top \nabla Q_i + O(\eta^2), \end{aligned}$$

provided the local second-order remainder is bounded. For a mixed-user batch, first-order effects add through its aggregate update. The controller tests that aggregate effect; it need not enumerate source–recipient pairs.

C Detailed Experimental Setup

C.1 Offline Recommendation Settings

The offline benchmark comprises two user-population scales from KuaiRand: KuaiRand-1K and KuaiRand-27K. Both contain sequential user–video interactions and associated user and item features. The KuaiSim-based environment uses these data to model user responses to recommended items. One decision selects one item; simulated click feedback supplies the immediate reward, and the configured continuation rule determines whether the session proceeds. The response model, reward definition, candidate-generation rule, and termination configuration are shared across methods within each setting.

Comparisons use DQN, A2C, DDPG, and TD3 as backbones. The method variants and grouping controls follow Section 5.1. Backbone-specific target construction and data requirements are preserved: A2C uses on-policy experience, while the replay-based algorithms reuse collected transitions.

C.2 Training and Evaluation Protocol

Each configuration uses three independent training seeds. Within a dataset–backbone comparison, methods have the same environment-interaction budget. Appendix C.3 documents the HAVL–DQN configuration and its fixed-budget checkpoint rule. The primary outcome is average cumulative reward as defined in Eq. (14). Evaluation assigns an equal number of complete sessions to each test user, computes a session-level average for each training seed, and reports the mean and sample standard deviation across the three seed-level values. The offline three-seed protocol is separate from the user-randomized online comparison described in Appendix C.4.

Grouping ablations replace only the grouping criterion while retaining the controller, group count, and reference-sampling protocol. Activity groups use interaction counts from the training data. The paired-update diagnostic uses an independent audit set that does not supply update constraints or hyperparameter-selection feedback.

Appendix C.3 specifies the data splits, response model, architecture, optimizer, grouping and controller settings, interaction budget, and evaluation configuration for HAVL–DQN. Appendix C.5 provides the diagnostic protocols, and Appendix D links the implementation.

C.3 Implementation Details

We describe the HAVL–DQN configuration on KuaiRand-27K with seed 1001. Tables 4–5 summarize its fixed settings.

C.3.1 Data, Response Model, and Session Evaluation

The processed 27K-source log contains 18,148,768 interactions from 13,524 users over 144,488 items. The response model uses a within-user sequence holdout: the final five records are reserved for testing, the preceding five for validation, and the remainder for training. The recorded split contains 18,013,528 training records and 67,620 records in each holdout set.

We reuse the pretrained `KRMBUserResponse` checkpoint and freeze it during policy learning. Its training log records one epoch of Adam with learning rate 10^{-4}, batch size 128, no L_2 penalty, and the supplied multi-feedback BCE objective. The model jointly predicts seven feedback types; the RL reward uses only sampled click feedback. User and item latent dimensions are 32, the encoding dimension is 64, and the history limit is 100. The sequence encoder has two Transformer layers, four attention heads, feed-forward width 64, and dropout 0.1. The checkpoint is saved after the single training epoch, without validation-based checkpoint selection.

Policy learning uses a separate user-level hash split with seed 1701 and thresholds 0.8 and 0.9, yielding 10,808 training users, 1,319 validation users, and 1,397 test users. This partition applies to policy learning and auxiliary validation, not to fitting the shared response model. Training starts are sampled from logged contexts within the training-user partition. Evaluation samples one starting context uniformly within each test user's history, giving 1,397 complete sessions for this configuration, and averages undiscounted cumulative reward equally across users.

Each decision recommends one item. The simulator initializes the remaining temper at $v_0 = 20$ and updates it as $v_{t+1} = v_t + r_t - 2$, where $r_t \in \{0, 1\}$ is the sampled click. A session terminates when $v_{t+1} < 1$. The item-correlation coefficient is 0.2. Evaluation uses a safety horizon of 25; any session that has not terminated by then raises an error rather than being discarded or treated as a completed session.

Table 4: Training and evaluation settings for the documented HAVL–DQN configuration on the KuaiRand-27K supported subset.

Setting	Value
Policy split (train / validation / test users)	10,808 / 1,319 / 1,397; hash seed 1701
Simulator response-model training	1 epoch; Adam, 10^{-4}; batch size 128
Maximum logged history length	100
Initial temper / evaluation safety horizon	20 / 25
Reward / training discount	Sampled click / $\gamma = 0.9$
State / item-feature dimensions	193 / 64
Critic hidden widths / activation	$[256, 128]$ / ReLU
Critic optimizer	Adam, 10^{-3}; weight decay 10^{-5}
TD batch size / replay capacity	128 / 150,000 per replay
Vectorized environments	32
Warm-up / subsequent policy interactions	12,800 / 100,000
Critic updates / target Polyak coefficient	3,125 / 0.01
Behavior / target candidate counts	512 / 512
Exploration probability	Linear decay from 1.0 to 0.1
Checkpoint	Final fixed-budget critic
Evaluation sessions	1 per test user; 1,397 in total
Evaluation candidate scoring	Full catalog; chunks of 8,192 items

C.3.2 Auxiliary Predictor and Grouping Updates

The state concatenates the frozen response model's 192-dimensional user encoding with normalized remaining temper, giving 193 dimensions; each item has a frozen 64-dimensional encoding. The feature map z projects the state to $q = 16$ dimensions using a fixed orthogonal matrix obtained by QR decomposition of a Gaussian matrix. Thus, the pretrained encoder supplies the features, while the final projection is fixed rather than learned jointly with the critic.

The auxiliary predictor receives the 257-dimensional state–item input. Its trunk has two width-128 ReLU layers followed by a 128-dimensional linear latent layer, with separate reward, continuation-logit, and 16-dimensional successor heads. Supervision is $(r, 1-d, \gamma(1-d)z(s'))$, with detached targets and parameters separate from the critic. Reward MSE, continuation BCE, and successor MSE receive equal component weights; successor MSE averages over its 16 coordinates, corresponding to $\lambda_m = 1/16$ when Eq. (5) is written as a squared Euclidean norm. The predictor is pretrained for 10 epochs on 10,240 non-reference warm-up transitions and evaluated on 2,560 separately collected validation transitions. Thereafter, it receives one Adam update per critic iteration, followed by a teacher update with Polyak coefficient 0.01.

Grouping uses $K = 8$ prototypes and temperature $T_g = 0.5$. At initialization and every 100 critic updates, 2,048 user-balanced reference contexts initialize prototypes by farthest-point selection in coordinate-standardized descriptor space. A further 2,048 reference draws calibrate the radius to the 90th percentile of the nearest-prototype distance. Between these refreshes, prototypes move toward their membership-weighted reference means with coefficient 0.02 after each control step. These moving centers are descriptor-space prototypes rather than necessarily observed contexts.

The implementation calibrates distance weights from the prototype-fitting descriptors. Let σ_r, σ_p, and $\sigma_{m,j}$ denote their coordinatewise sample standard deviations, each floored at 10^{-4}. The implemented distance is

$$D_{\mathrm{impl}}(i,j) = a_r|\hat{r}_i - \hat{r}_j| + a_p\gamma|\hat{p}_i - \hat{p}_j| + a_m\|\hat{m}_i - \hat{m}_j\|_2,$$
$$a_r = \sigma_r^{-1}, \qquad a_p = \sigma_p^{-1}, \qquad a_m = \left(\frac{1}{q}\sum_{j=1}^{q}\sigma_{m,j}^2\right)^{-1/2}. \tag{22}$$

This is Eq. (6) up to the common factor a_r, with $C_0 = a_p/a_r$ and $C_z = a_m/a_r$. Dividing the distance, radius, and temperature by a_r preserves all memberships. The radius and scales are recalibrated at each prototype refresh and held fixed between refreshes; the teacher, prototypes, memberships, and TD labels remain fixed throughout each critic projection and acceptance check. Contexts outside all radii retain zero reference membership but still contribute to ordinary TD learning.

C.3.3 Critic Training and Transfer Constraints

The critic is a state–item MLP with ReLU hidden widths 256 and 128 and a scalar output. Training collects 12,800 random warm-up transitions, followed by 100,000 policy-generated transitions over 3,125 vectorized steps of 32 environments. One in every five transition identifiers is assigned to the separate reference replay; the remainder supplies ordinary TD batches. The critic minimizes mean squared TD error on transition-uniform batches of 128, using Adam with learning rate 10^{-3} and weight decay 10^{-5}. The target-network Polyak coefficient is 0.01, and the final fixed-budget critic is evaluated.

For the large item catalog, behavior and TD-target maximization each use 512 uniformly sampled candidate indices. Behavior exploration decreases linearly from 1.0 to 0.1, with random actions drawn from the catalog. Evaluation instead scans all items in chunks of 8,192 and masks items in the current history. This candidate-based training maximization is the DQN adaptation used here; it is not an exhaustive training-time maximization over the catalog.

Every critic update draws 512 user-balanced reference contexts. Only groups with effective sample size $(\sum_j \omega_{jk})^2/\sum_j \omega_{jk}^2 \geq 16$ enter the constraints. Using the half-squared reference losses in Eq. (10), the nonlinear allowance, linear budget, and trust radius are

$$\tau_k = 0.01\max\{L_k, 0.1\}, \qquad b_k = \tau_k/2, \qquad \rho = \|\Delta_0\|_2. \tag{23}$$

Table 5: HAVL grouping and transfer-control settings. The radius and distance scales are calibrated from reference descriptors rather than fixed as absolute distances.

Setting	Value or calibration rule
Number of groups K / feature dimension q	8 / 16
Auxiliary pretraining	10 epochs; batch size 256; Adam, 10^{-3}
Auxiliary component loss weights	$(1, 1, 1)$; coordinate-mean successor MSE
Online auxiliary updates	1 per critic iteration; batch size 128
Auxiliary teacher coefficient / frequency	0.01 / every critic iteration
Prototype-fitting / calibration draws	2,048 / 2,048
Grouping radius	90th percentile of nearest-prototype distance
Soft-membership temperature T_g	0.5 in implemented distance units
Distance weights	Inverse empirical scales; Eq. (22)
Prototype refresh / center update	Every 100 iterations / EMA coefficient 0.02
Reference allocation / batch size	Every fifth transition / 512
Reference sampling	Uniform users, then uniform within-user transitions
Minimum group effective sample size	16
Control frequency	Every critic update
Allowed nonlinear loss increase τ_k	$0.01 \max\{L_k, 0.1\}$
Linear budget b_k / trust radius ρ	$\tau_k/2$ / $\|\Delta_0\|_2$
Backtracking	Factor 0.5; initial trial plus 10 halvings
Nonlinear-check numerical tolerance	10^{-7}
Projection tolerance / sweep limit	10^{-8} / 1,000

The controller acts on all critic parameters. It tests $\alpha \in \{1, 2^{-1}, \ldots, 2^{-10}\}$ against the fixed reference objectives, allowing an additional numerical tolerance of 10^{-7}. Failure of all 11 trials skips the update. If no group meets the effective-sample-size threshold, the ordinary optimizer update is applied.

C.3.4 Constrained Update Solver

Let A have rows $a_k^\top$ and introduce multipliers $\lambda \geq 0$ for $A\Delta \leq b$ and $\nu \geq 0$ for $(\|\Delta\|^2 - \rho^2)/2 \leq 0$. The Lagrangian is

$$\tfrac{1}{2}\|\Delta - \Delta_0\|^2 + \lambda^\top(A\Delta - b) + \tfrac{\nu}{2}(\|\Delta\|^2 - \rho^2).$$

Its unconstrained minimizer is $\Delta = (\Delta_0 - A^\top\lambda)/(1 + \nu)$. Substitution yields the concave dual objective

$$g(\lambda, \nu) = \tfrac{1}{2}\|\Delta_0\|^2 - \frac{\|\Delta_0 - A^\top\lambda\|^2}{2(1 + \nu)} - \lambda^\top b - \tfrac{\nu}{2}\rho^2. \tag{24}$$

The squared term depends only on $AA^\top$, $A\Delta_0$, and $\|\Delta_0\|^2$. The implementation uses nonnegative coordinate updates for the group multipliers, with tolerance 10^{-8} and at most 1,000 sweeps. If the trust constraint is active, its multiplier is bracketed by at most 80 doublings and refined by 60 bisection iterations. Strictly positive budgets and radius admit Slater's condition using $\Delta = 0$, giving strong duality; $\rho = 0$ returns zero directly. The solver records primal residuals and a convergence flag, and the nonlinear acceptance check is applied to the returned candidate.

C.3.5 Optimizer State and Computational Cost

The controller first computes Adam's proposed parameter update, including weight decay, and restores the original parameters before projection. It then applies the accepted corrected update directly, without a second optimizer step. Provisional Adam moments are retained for a nonzero accepted update and restored when the step is skipped. The target critic and auxiliary teacher update after the critic decision. The critic contains neither dropout nor batch normalization, so reference forwards do not change stochastic masks or running statistics during backtracking.

For P critic parameters and K_t active reference groups, each control step requires one proposal gradient and K_t reference gradients, with direct storage cost $O(K_tP)$. The dual solver uses a $K_t \times K_t$ Gram matrix; backtracking adds reference forward passes. Inference uses the shared critic and candidate generator without the auxiliary predictor or controller.

C.4 Online Experiment Scenario

The online application concerns rewarded advertising in short-drama services delivered through Tencent's WeChat Mini Programs. Users voluntarily watch an advertisement to unlock additional episodes. As illustrated in Figure 3, an ad request invokes the decision engine, which selects the ad duration and reward level. After ad completion, the reward mechanism grants access to the selected amount of content. Further viewing can lead to another ad request, producing a sequence of decisions within the same user session.

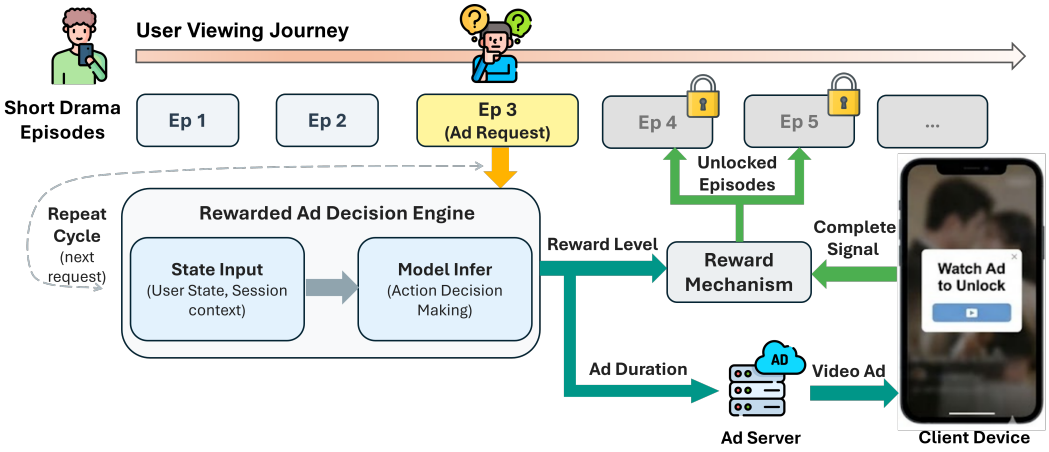


Figure 3: Rewarded-ad delivery in a WeChat Mini Program short-drama service. The decision engine uses the user state and session context to select ad duration and content-unlocking reward. Completing the ad unlocks subsequent episodes, and later requests repeat the decision cycle.

The state includes user-profile features, recent content interactions, and the current session context. An action is a rewarded-ad configuration rather than an item identifier. Its reward level determines the amount of content unlocked and affects the interval before the next ad request. The learning reward combines advertising monetization and user experience, while the value function estimates the cumulative return across subsequent decisions. This application retains the shared, user-conditioned value-learning problem while changing the action space from content selection to ad configuration.

The A/B test ran for 7 days, comparing the production DQN policy with the same backbone trained using HAVL, with assignment at the user level. CTR and GMV serve as business outcomes; relative changes are computed against the corresponding control-group metrics. The intervention concerns the training of the shared value function, while the serving model continues to select an ad configuration from the available actions.

C.5 Cross-User Sharing Diagnostics

Distance-ranking protocol. The diagnostic samples two contexts per user for 4,096 users and constructs 200,000 cross-user pairs with exactly the same current action. A shared pool of 64 items supplies the current actions. Each context receives 32 repeated one-step simulator draws. For eight fixed continuation functions, we estimate

$$\widehat{B}_\ell(x) = \frac{1}{32}\sum_{b=1}^{32}\left[r^{(b)} + \gamma(1-d^{(b)})V_\ell(s'^{(b)})\right], \qquad \gamma = 0.9.$$

Four continuation functions evaluate a frozen critic at four fixed probe actions; four are random nonlinear functions of the next-state features. The evaluation discrepancy is $\frac{1}{8}\sum_{\ell=1}^{8}|\widehat{B}_\ell(x_i) - \widehat{B}_\ell(x_j)|$. It is distinct from the distance used to select pairs, but the critic-based probes are not independent of the critic representation being evaluated. All distances rank the same candidate-pair pool. Each curve reports the mean discrepancy among the closest 1, 5, 10, 20, 50, and 100% of pairs under its ranking; at full coverage, every ranking includes the same pairs.

Scalar-value distance is the absolute difference between predictions. State and critic-representation rankings use cosine distances; the descriptor diagnostic uses a standardized Euclidean distance over predicted reward, continuation probability, and successor features. This cache evaluates descriptor-based pair selection, not the weighted distance and prototype memberships in Section 4.1. Finite simulator sampling introduces uncertainty in the estimated backup responses; the displayed curve is not a confidence bound on expected-backup compatibility.

Paired-update protocol. The audit uses one fitted DQN training seed at checkpoint 150 and 32 non-overlapping proposal batches of 128 transitions. A separate reference pool contains 1,024 transitions. The episode-disjoint audit pool initially contains 1,942 transitions from 134 users; excluding proposal-batch users leaves 129–134 receiver users per update. Audit episodes did not participate in critic training, descriptor fitting, grouping or radius selection, reference constraints, or acceptance checks. Each branch starts from the same learner snapshot, uses the same frozen target labels, and is evaluated before the snapshot is restored for the next proposal.

For an audit transition, let $e_i(\Delta) = \ell_i(\theta + \Delta) - \ell_i(\theta)$. Cross-user harm is measured by

$$H(\Delta) = \mathbb{E}_i[\max(e_i(\Delta), 0)].$$

Positive loss changes are computed per transition, averaged within each receiver user, then averaged equally over users and proposal batches. With a relative group-loss tolerance of 1%, HAVL modifies 84.4% of proposals and skips none.

Table 6: Cross-user harm in the paired-update audit. H measures mean positive fixed-target TD-loss change. Results are averaged over 32 proposals from one DQN checkpoint.

Update	$H \downarrow$
Base	0.006139
HAVL	0.003308

As shown in Table 6, HAVL reduces H by 46.1% relative to Base. The 95% bootstrap intervals in Figure 2(b) resample proposal batches and condition on the fitted model and audit pool; they do not measure variability across training seeds or guarantee a reduction in true value-estimation error.

D Source Code

The source code is available at https://anonymous.4open.science/r/HAVL-C560/.