RECOMMENDATION WORLD MODELS FOR FUTURE-STATE CONTROL

Anonymous authors
Paper under double-blind review

ABSTRACT

Sequential recommendation optimizes which items to rank, while each displayed slate also shapes subsequent feedback and user state. We study how a trained ranker can support decisions about these future consequences. We introduce UA-TWM, a utility-anchored world-model interface that constructs nearby slate actions, estimates their target-relevant consequences, and selects an alternative subject to utility constraints. The reference slate serves as a fallback when no alternative qualifies. A logged-replay instantiation combines utility and target-gain estimates with calibrated failure-risk prediction; a closed-loop instantiation uses one-step state-action prediction and updates its decisions after observed feedback. We evaluate transfer across twelve sequential backbones on MovieLens-25M and KuaiRand-Pure, and repeated target-directed interaction in KuaiSim. Attaching the interface improves Recall@20, NDCG@20, and future-state alignment for every matched logged backbone. Selection ablations reveal the utility and risk costs of aggressive target pursuit, while closed-loop diagnostics isolate the contribution of action-conditioned prediction. Local consequence modeling thus enables target-aware selection around a trained sequential ranker. Code can be found at https://anonymous.4open.science/r/UA-TWM-0CD6.

1 INTRODUCTION

Sequential recommenders have become strong ranking models. Given a user history, they score the next item and return a slate with high short-term utility (Hidasi et al., 2016; Xu et al., 2025a; Sun et al., 2019; Zhou et al., 2020; Tan et al., 2021; Xu et al., 2026a). Deployment adds a different question. A displayed slate changes the feedback available to the system and the user state used by the next recommendation step. Two slates can be close in Recall or NDCG while inducing different future category mixtures, repeat exposure, or interest concentration. A next-item score therefore does not, by itself, order the future consequences of candidate slate actions.

The difference is easiest to see at the slate level. Suppose two candidate slates both contain items that are plausible next clicks for the same user. One slate may concentrate the next session around a narrow genre, while the other keeps the same expected utility but broadens the user's subsequent consumption mixture. Choosing between these slates requires estimates of their future effects and the utility cost of steering toward a target. Item scores identify relevant candidates; evaluating their consequences requires conditioning on the slate as a whole and on the current user state.

This gap is central for recommendation systems that need to reason about future-state statistics rather than a single next click. A platform may want to preserve utility while reducing concentration, increasing exposure balance, or moving a user toward a requested content mixture. These targets are properties of trajectories produced by slate actions and feedback, not labels on isolated items. Interactive simulators and long-horizon recommendation methods make this feedback loop explicit (Ie et al., 2019a; Xu et al., 2026b; Shi et al., 2019; Zhao et al., 2023; Gao et al., 2023b;a; Xu et al., 2025b), but most sequential rankers expose only a local ranking surface. A decision layer around a strong ranker can evaluate these consequences locally: it compares plausible slates by expected utility, target movement, and estimated failure risk.

List diversity and calibration provide useful ways to construct candidate slates. Consequence-aware selection adds a temporal question: which of these candidates is predicted to move the future statistic while preserving utility? The ranking model supplies a relevance-based reference, candidate

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.

generation exposes alternative actions, and consequence prediction evaluates their effects. This decomposition allows the decision layer to use the structure learned by a sequential ranker while optimizing an explicit target beyond next-item relevance.

We introduce UA-TWM, a utility-anchored recommendation world-model interface. A frozen recommender supplies the anchor slate, and a local consequence layer scores alternative slates against that reference. In logged replay, the interface combines utility and target-gain estimates with a calibrated failure-risk predictor. In closed-loop control, a learned state-action model predicts one-step consequences, and the controller selects a target-improving slate subject to a utility check. The anchor provides a fallback when no candidate passes selection. After a non-anchor selection, observed feedback determines whether the predicted target improvement and utility conditions hold.

We evaluate transfer and interaction in complementary settings. MovieLens-25M and KuaiRand-Pure provide chronological logs for matched comparisons across twelve sequential backbones. KuaiSim provides repeated interaction: the controller predicts one step ahead, executes a slate, observes feedback, and updates its state over ten steps. The logged comparisons measure alignment with historical future behavior; the simulator measures target movement following executed slates. Each simulated decision uses a one-step prediction within the ten-step interaction.

Our contributions are threefold. First, we formulate future-state-aware recommendation as selective slate-action control with anchor-relative utility and target criteria. Second, we instantiate this formulation as a local consequence interface that separates candidate generation, prediction, and constrained selection. Third, we evaluate both cross-backbone transfer and closed-loop behavior: matched logged comparisons improve Recall@20, NDCG@20, and future-state alignment across two public domains, while action-conditioning and selection ablations examine the mechanisms behind those decisions. This approach reuses the ranker's relevance estimates while learning how candidate actions affect the quantities needed for target-directed selection.

2 Related Work

Sequential recommendation as utility anchors. Sequential recommenders learn user-history representations for next-item ranking. Recurrent, convolutional, self-attentive, bidirectional, transition-based, self-supervised, multi-interest, diffusion, and flow-based models define heterogeneous ranking functions (Hidasi et al., 2016; Tang & Wang, 2018; Kang & McAuley, 2018; Sun et al., 2019; Yuan et al., 2019; Rendle et al., 2010; He et al., 2017; Liu et al., 2018; Fan et al., 2021; Zhou et al., 2020; Tan et al., 2021; Chen et al., 2025; Shi et al., 2026). These functions supply the relevance scores and reference slates for our local action search. Matched evaluations hold each backbone fixed and measure the effect of consequence-aware selection around it.

Controllable and interactive recommendation. Goal-conditioned recommendation and reinforcement-learning approaches make desired outcomes or long-horizon value explicit (Wang et al., 2021; Li et al., 2024; Gao et al., 2025; Ie et al., 2019b; Zhao et al., 2018; Bai et al., 2019; Gao et al., 2023b;a). Simulators and randomized recommendation logs provide complementary resources for studying user feedback (Ie et al., 2019a; Shi et al., 2019; Gao et al., 2022; Zhao et al., 2023). Our controller searches candidate slates near the ranker's reference recommendation, evaluates their target movement and utility cost, and retains that reference when the selection criteria are unmet.

World models for selective action. World models learn dynamics for prediction, planning, or control (Sutton, 1991; Ha & Schmidhuber, 2018; Chua et al., 2018; Janner et al., 2019; Hafner et al., 2020; 2021; Schrittwieser et al., 2020). Return-conditioned sequence modeling provides another route to decision making (Chen et al., 2021). In recommendation, the state summarizes user history, the action is a slate, and the predicted quantities describe subsequent feedback and consumption. Recent LLM-based user simulators and judges address related evaluation problems (Bonin et al., 2025; Bougie et al., 2026). Our closed-loop instantiation uses one-step state-action prediction for repeated constrained selection. Appendix Section D expands the comparison with related recommendation approaches.

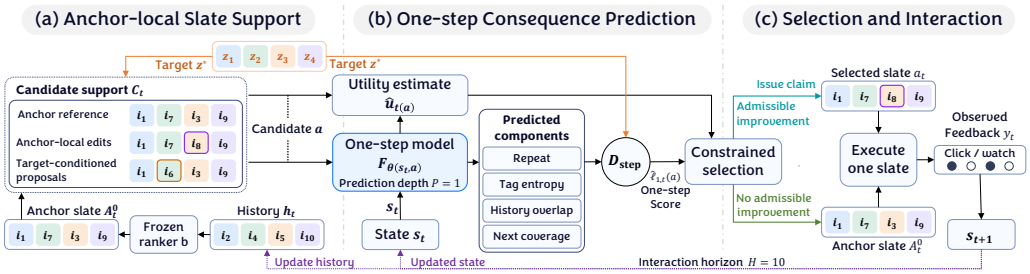

Figure 1: Closed-loop UA-TWM: (a) anchor-local slate proposals, (b) one-step consequence prediction, and (c) constrained selection with feedback. Prediction depth is $P = 1$; interaction lasts $H = 10$ steps. Slates are schematic.

3 Methodology

3.1 Decision Problem and Evidence Regimes

We model repeated recommendation as a controlled process. At time t, the system summarizes the history $h_t = (a_0, y_0, \ldots, a_{t-1}, y_{t-1})$ into a state $s_t = g_\psi(h_t)$, selects a slate $a_t = (i_{t,1}, \ldots, i_{t,K})$, and observes feedback y_t. The environment induces

$$P^\star(s_{t+1}, y_t \mid s_t, a_t) = P^\star(y_t \mid s_t, a_t) P^\star(s_{t+1} \mid s_t, a_t, y_t). \tag{1}$$

A target map ϕ extracts a horizon-H future-state statistic from the induced trajectory, such as a category mixture, concentration score, or repeat exposure statistic. For a target $z^\star$, the interventional target loss and horizon utility of a policy π are

$$L_H(\pi; s, z^\star) = \mathbb{E}_{P^\star,\pi}\left[D(\phi(\tau_{t:t+H}), z^\star)\right], \quad V_H(\pi; s, z^\star) = \mathbb{E}_{P^\star,\pi}\left[\sum_{h=0}^{H-1} r(y_{t+h}, a_{t+h})\right]. \tag{2}$$

The utility-safe control problem is to reduce L_H without dropping below a strong recommendation anchor π_{anc}:

$$\min_\pi \mathbb{E}_{(s,z^\star)}[L_H(\pi; s, z^\star)] \quad \text{s.t.} \quad \mathbb{E}_{(s,z^\star)}[V_H(\pi; s, z^\star)] \geq \mathbb{E}_{(s,z^\star)}[V_H(\pi_{\text{anc}}; s, z^\star)] - \epsilon_u. \tag{3}$$

We approximate this control problem through local action-conditioned prediction. Given the current state, a target, and candidate slates around the anchor, the consequence model estimates target movement and the selector checks utility preservation. The model scores the candidate slates available to the controller. The policy selects a qualifying alternative or returns the anchor.

Closed-loop rollout evaluates the induced trajectory under the selected policy. Logged replay evaluates a retrospective proxy: prefix-generated candidate slates are compared with held-out future positives collected under the historical policy. The former measures action-induced target movement within the simulator; the latter measures future-statistic alignment under chronological replay. Logged positives do not identify the transition distribution of an unshown slate.

Both regimes evaluate utility and target conditions relative to the same anchor. In rollout, failure concerns induced utility and trajectory statistics. In replay, failure concerns ranking utility and alignment with the historical future-positive mixture. Coverage records non-anchor selection frequency, and invalidity records failures among those selections.

3.2 UA-TWM: A Utility-Anchored World-Model Interface

UA-TWM converts a trained sequential recommender into a selective slate-action controller. The backbone is frozen and supplies the base slate $A_{u,t}^0$. The consequence layer is trained only around candidate slates near this anchor. It predicts utility gain, future-state gain, and support risk, then uses these predictions to decide whether to select a non-anchor slate or return the anchor.

Candidate support. Let $c_i \in \{0,1\}^d$ denote the observable multi-hot genre or tag vector of item i, and let $\mathcal{P}_M(h)$ be the top-M pool returned by the frozen backbone after masking items already seen in the prefix. Profiles are normalized after summing item vectors. The decision-time target is $z^\star$ in a controlled rollout and $z^\dagger_{u,t}$ in logged replay. The logged target is prefix-only:

$$z^\dagger_{u,t} = \frac{\sum_{i \in \mathcal{H}^{+,\text{train}}_{u,t}} c_i + \omega \sum_{i \in \mathcal{H}^{+,\text{calib}}_{u,t}} c_i}{\left\| \sum_{i \in \mathcal{H}^{+,\text{train}}_{u,t}} c_i + \omega \sum_{i \in \mathcal{H}^{+,\text{calib}}_{u,t}} c_i \right\|_1}, \tag{4}$$

with $\omega = 2$ weighting the more recent calibration block. Calibration scoring uses training history; test scoring uses the completed training and calibration blocks. An empty profile returns the zero vector. Appendix A.1 specifies the decision boundaries, normalization, and future windows. The steering score used for target-conditioned proposals is

$$q(i, z, h) = \langle c_i, z \rangle, \tag{5}$$

so the target signal can change the order of plausible backbone candidates without introducing items outside the anchor's local support. Logged candidates are generated by

$$A^\beta(h, z) = \text{TopK}_{i \in \mathcal{P}_M(h)} \left[\tilde{s}_b(i \mid h) + \beta q(i, z, h) \right], \qquad \beta \in \mathcal{B}. \tag{6}$$

Here $\tilde{s}_b$ is the within-pool standardized backbone score, and β is selected on calibration users. The closed-loop instantiation constructs candidates from two proposal sources. Anchor-local edits keep most items from A^0 and replace a small number of low-contribution items with high-utility candidates whose item/tag summaries improve the requested statistic. Graph-semantic replacements are fixed proposal generators over observable state and item metadata. They form candidate pools from immediate click scores, history-repeat penalties, history-tag similarity penalties, and within-slate diversity penalties; then they instantiate repeat-oriented, balanced, and diversity-oriented slates under preset click-floor variants. Both proposal sources use the state and item metadata available before execution; the consequence model then scores the resulting slates. In the KuaiSim diagnostic, each decision has nine candidate actions: the anchor, two anchor-local edits, and six graph-semantic replacement slates. Candidate construction defines the controller's local search space around the utility anchor.

Replay consequence heads. Let $\mathcal{F}^+_{u,t}$ contain the positive events in the held-out block following the decision boundary. The logged future statistic is

$$\tilde{z}_{u,t} = \phi(\mathcal{F}^+_{u,t}) \tag{7}$$

and uses the full variable-length block to supply calibration labels or test outcomes. The H-step horizon applies to closed-loop interaction. A candidate slate A is scored by

$$U(A, u) = \text{Recall@20}(A, \mathcal{F}^+_{u,t}), \qquad S(A, u) = \|\phi(A) - \tilde{z}_{u,t}\|_1. \tag{8}$$

The consequence labels compare each candidate with the anchor:

$$\Delta_U(u, A) = U(A, u) - U(A^0, u), \qquad \Delta_Z(u, A) = S(A^0, u) - S(A, u). \tag{9}$$

The logged implementation estimates target and utility margins from prefix statistics and backbone scores. The predicted state movement is the prefix-target margin

$$\hat{g}_Z(u, A) = \|\phi(A^0) - z^\dagger_{u,t}\|_1 - \|\phi(A) - z^\dagger_{u,t}\|_1, \tag{10}$$

and the predicted utility margin is the mean standardized backbone-score margin between A and A^0:

$$\hat{g}_U(u, A) = \frac{1}{K} \sum_{i \in A} \tilde{s}_b(i \mid h_t) - \frac{1}{K} \sum_{i \in A^0} \tilde{s}_b(i \mid h_t). \tag{11}$$

The risk head is learned from calibration candidates. Its feature vector contains only prefix- and candidate-support quantities: β, $\hat{g}_Z$, $\hat{g}_U$, target concentration $\max_j z^\dagger_{u,t,j}$, $\hat{g}_Z \max_j z^\dagger_{u,t,j}$, $\beta \hat{g}_Z$, $\hat{g}_U \max_j z^\dagger_{u,t,j}$, $\max(0, -\hat{g}_U)$, $\hat{g}_Z^2$, and $\hat{g}_U^2$. The held-out statistic $\tilde{z}_{u,t}$ is not part of $x(u, A)$. The consequence interface therefore has three action-conditioned outputs,

$$\hat{g}_U(u, A), \qquad \hat{g}_Z(u, A), \qquad \hat{r}(u, A), \tag{12}$$

where $\hat{r}$ predicts failure of either the target-gain margin or the per-user utility floor:

$$y_{\text{risk}}(u, A) = \mathbf{1}[\Delta_Z(u, A) < \delta_Z \ \vee \ \Delta_U(u, A) < -\epsilon_{\text{user}}]. \tag{13}$$

The reported logged runs train $\hat{r}$ as an L_2-regularized logistic classifier with binary cross entropy,

$$\mathcal{L}_{\text{risk}} = \text{BCE}(\hat{r}, y_{\text{risk}}) + \lambda \|\theta_r\|_2^2. \tag{14}$$

For the split-risk diagnostic, the binary label is decomposed into $y_U = \mathbf{1}[\Delta_U < -\epsilon_{\text{user}}]$ and $y_Z = \mathbf{1}[\Delta_Z < \delta_Z]$, yielding two probabilities $\hat{r}_U$ and $\hat{r}_Z$ whose maximum defines the gate risk. Thresholds are chosen on calibration users. Test future positives never enter candidate generation or threshold selection.

Selective planning and claims. Let $\mathcal{C}_{\text{adm}}(u) = \{A \in \mathcal{C}(u) : \hat{g}_U(u, A) \geq \tau_U, \ \hat{g}_Z(u, A) \geq \tau_Z, \ \hat{r}(u, A) \leq \rho\}$ be the candidates that pass all three gates. At test time the logged selector returns

$$\pi_{\text{UA}}(u) = \begin{cases} \arg\max_{A \in \mathcal{C}_{\text{adm}}(u)} \hat{g}_Z(u, A), & \mathcal{C}_{\text{adm}}(u) \neq \varnothing, \\ A^0_{u,t}, & \text{if no candidate satisfies the gates.} \end{cases} \tag{15}$$

For closed-loop evaluation, a non-anchor selection constitutes a target-control claim made before feedback is observed; anchor retention issues no claim. For a selected non-anchor action a at state s and target $z^\star$, the claim is valid when the induced rollout improves the target statistic while preserving the anchor utility:

$$\Gamma^\star(s, z^\star, a) = \mathbf{1}\left[L_H(a; s, z^\star) \leq L_H(a_{\text{anc}}; s, z^\star) - \delta_z \wedge V_H(a; s, z^\star) \geq V_H(a_{\text{anc}}; s, z^\star) - \epsilon_u\right]. \tag{16}$$

Coverage is the fraction of evaluated scenarios with an issued non-anchor claim. Invalidity is the fraction of issued claims that fail Equation 16; unsuccessful claims remain in the denominator. Returning the anchor contributes to neither the claim count nor the invalid-claim count.

Calibration and operating points. The thresholds (τ_U, τ_Z, ρ) define the logged selector's operating point and are chosen on calibration users before test evaluation. The utility threshold limits the predicted utility cost, the target threshold sets the required predicted movement, and the risk threshold filters candidates by estimated failure. For fixed utility and target thresholds, a tighter risk gate retains fewer candidates. Selection maximizes predicted target gain over those that remain.

The ablations isolate these decisions. Removing the utility floor tests the cost of target movement; removing the state head tests the contribution of target prediction; removing the risk gate tests whether gain estimates suffice to select reliable candidates. Removing fallback forces a non-anchor choice when the full selector would abstain, exposing the cost of intervention on those cases.

One-step prediction and repeated interaction. The closed-loop controller fits a ridge state-action model to offline tuples of aggregate states, slate summaries, feedback, and next-state statistics:

$$\min_{\theta} \sum_{(s,a,y,s') \in \mathcal{D}_{\text{dyn}}} \|F_\theta(s, a) - \psi(y, s')\|_2^2 + \lambda_{\text{dyn}} \|\theta\|_2^2, \tag{17}$$

Here λ_{dyn} is the ridge regularization coefficient. The prediction depth is $P = 1$ and the interaction horizon is $H = 10$. A readout q_1 extracts predicted repeat exposure, tag entropy, history-tag overlap, and next-state coverage. The one-step target score and anchor-relative gain are

$$\hat{z}_t^{(1)}(a) = q_1(F_\theta(s_t, a)), \qquad \hat{\ell}_{1,t}(a) = D_{\text{step}}\left(\hat{z}_t^{(1)}(a), z^\star\right), \tag{18}$$

$$\hat{g}_{Z,t}^{(1)}(a) = \hat{\ell}_{1,t}(a_t^0) - \hat{\ell}_{1,t}(a), \tag{19}$$

where D_{step} compares predicted components with the target. The controller minimizes $\hat{\ell}_{1,t}$ among admissible candidates, using an immediate click estimate for the utility check, and retains the anchor unless its target score improves. After executing one slate, observed feedback updates the state and candidates are regenerated. This repeats for ten environment steps. The trajectory-level L_H and V_H evaluate the resulting interaction; selection uses one-step predictions throughout.

4 EXPERIMENTS

4.1 EVALUATION PROTOCOL

The experiments examine three questions: does the consequence interface improve recommendations across backbones, which selection components account for the improvement, and does action-conditioned prediction support target-directed interaction? We use chronological replay for the cross-backbone comparison and KuaiSim rollouts for the interaction test. Alternatives and ablations share the candidate support of the corresponding full method.

MovieLens-25M uses a frozen chronological 70/10/20 split over 162,541 users and 25M ratings. Utility is Recall@20 and NDCG@20 over future high-rated items. Future-state loss is the L_1 distance between the genre mixture of the deployed slate and the genre mixture of future high-rated items. KuaiRand-Pure uses a chronological short-video split over 17,269 users, 7,583 videos, and 111 tag-state categories. A positive interaction denotes click or long view. Future-state loss is the L_1 distance between the deployed slate's tag mixture and the future positive-video tag mixture. Within each dataset, all methods share the same histories, item vocabulary, top-200 candidate pool, calibration split, slate size, and target-state definition. Logged replay uses $K = 20$, maximum history length 100, candidate weights $\beta \in \{0.05, 0.1, 0.2, 0.3, 0.5, 0.75, 1.0\}$, and calibration-selected utility, state, and risk gates. Closed-loop rollout uses prediction depth $P = 1$, interaction horizon $H = 10$, and slate size 6.

At each logged decision boundary, the target profile uses the available positive history, and the selected slate is evaluated against the following held-out positive block. This block has variable length across users. The profile weights each calibration-block event by $\omega = 2$ at test time; calibration scoring uses the training block alone. Item tags are summed before normalization. Appendix A.1 gives the event-level construction.

The logged protocol evaluates twelve sequential recommendation anchors grouped by model family. The canonical sequence encoders are GRU4Rec, SASRec, BERT4Rec, and NextItNet, which represent recurrent, self-attentive, bidirectional, and convolutional next-item modeling (Hidasi et al., 2016; Kang & McAuley, 2018; Sun et al., 2019; Yuan et al., 2019). The second group covers structurally distinct ranking surfaces: FPMC and TransRec supply Markov and metric-transition anchors, STAMP supplies a session-attention anchor, LightSANs and S3Rec supply attention and self-supervised sequence anchors, and SINE supplies a multi-interest anchor (Rendle et al., 2010; He et al., 2017; Liu et al., 2018; Fan et al., 2021; Zhou et al., 2020; Tan et al., 2021). The third group adds recent generative sequential recommenders: ADRec uses diffusion-style sequence prediction and FAVE uses flow-based average-velocity prediction (Chen et al., 2025; Shi et al., 2026). For external repositories, we convert the same frozen histories into each required training format and evaluate ranked slates with the same replay evaluator used for all anchors. For every backbone, Base is the trained recommender and +UA-TWM is the same recommender after attaching the consequence layer and constrained selector. The matched with/without design isolates the decision interface because the backbone parameters are unchanged.

We report ranking utility, future-state loss, coverage, and invalidity. Recall@20 and NDCG@20 measure retrieval and rank-sensitive utility, respectively. Future-state loss measures discrepancy from the regime-specific target statistic. Logged coverage counts non-anchor selections over evaluated users; closed-loop coverage counts issued target-scenario claims. Invalidity is conditional on issuing a claim. We interpret coverage together with invalidity because anchor retention can reduce failures without improving target control.

4.2 LOGGED TRANSFER ACROSS BACKBONES

Table 1 reports the matched logged test. Across two recommendation domains and twelve backbone families, attaching UA-TWM improves Recall@20, NDCG@20, and future-state alignment for every matched backbone. On MovieLens-25M, Recall@20 gains range from 0.0119 to 0.0144, NDCG@20 gains range from 0.0028 to 0.0102, and future-state L_1 drops by 0.037 to 0.046. On KuaiRand-Pure, Recall@20 gains range from 0.0054 to 0.0068, NDCG@20 gains range from 0.0018 to 0.0037, and future-state loss drops by 0.064 to 0.082. In relative terms, the median improvements are 5.9% Recall@20 and 5.9% NDCG@20 on MovieLens-25M, and 6.4% Recall@20 and 6.5% NDCG@20 on KuaiRand-Pure; the mean relative gains are larger because ADRec starts

Table 1: Logged transfer across sequential recommendation backbones. Attaching UA-TWM improves ranking utility and future-state alignment for every matched backbone on both datasets. R@20 is Recall@20, N@20 is NDCG@20, and S is future-state L_1 loss. I/C reports replay-unsafe rate / coverage in percent for non-anchor UA-TWM claims.

		MovieLens-25M				KuaiRand-Pure			
Backbone	Policy	R@20 ↑	N@20 ↑	S ↓	I/C	R@20 ↑	N@20 ↑	S ↓	I/C
Canonical sequence encoders									
GRU4Rec	Base	0.1177	0.0338	0.881	–	0.0645	0.0263	1.461	–
	+UA-TWM	**0.1304**	**0.0375**	**0.840**	3.4/56.2	**0.0703**	**0.0287**	**1.389**	5.1/95.7
SASRec	Base	0.1067	0.0371	0.886	–	0.0656	0.0257	1.453	–
	+UA-TWM	**0.1194**	**0.0415**	**0.845**	3.6/54.7	**0.0714**	**0.0280**	**1.381**	5.2/96.0
BERT4Rec	Base	0.2454	0.0989	0.749	–	0.1080	0.0389	1.392	–
	+UA-TWM	**0.2581**	**0.1040**	**0.708**	5.1/94.2	**0.1138**	**0.0410**	**1.320**	6.0/98.2
NextItNet	Base	0.1194	0.0382	0.878	–	0.0652	0.0256	1.455	–
	+UA-TWM	**0.1321**	**0.0423**	**0.837**	3.5/55.8	**0.0710**	**0.0279**	**1.383**	5.0/95.5
Transition, session, attention, and multi-interest anchors									
FPMC	Base	0.2184	0.0534	0.757	–	0.1036	0.0352	1.398	–
	+UA-TWM	**0.2328**	**0.0569**	**0.711**	3.5/57.8	**0.1104**	**0.0375**	**1.319**	5.0/98.9
TransRec	Base	0.2127	0.0451	0.758	–	0.0925	0.0311	1.426	–
	+UA-TWM	**0.2257**	**0.0479**	**0.716**	3.8/53.1	**0.0983**	**0.0331**	**1.344**	5.5/95.2
STAMP	Base	0.2216	0.0549	0.755	–	0.1024	0.0354	1.397	–
	+UA-TWM	**0.2341**	**0.0580**	**0.718**	4.1/52.4	**0.1081**	**0.0374**	**1.333**	5.9/96.1
LightSANs	Base	0.2441	0.0959	0.749	–	0.1029	0.0364	1.396	–
	+UA-TWM	**0.2560**	**0.1006**	**0.709**	5.2/93.8	**0.1084**	**0.0383**	**1.322**	6.1/97.4
S3Rec	Base	0.2492	0.0952	0.749	–	0.1148	0.0392	1.368	–
	+UA-TWM	**0.2612**	**0.0998**	**0.708**	5.1/94.4	**0.1202**	**0.0410**	**1.304**	6.0/98.6
SINE	Base	0.2199	0.0556	0.757	–	0.1045	0.0358	1.397	–
	+UA-TWM	**0.2320**	**0.0587**	**0.719**	3.3/63.5	**0.1100**	**0.0377**	**1.329**	5.6/99.0
Recent generative sequence backbones									
ADRec	Base	0.0245	0.0191	0.819	–	0.0591	0.0368	1.430	–
	+UA-TWM	**0.0372**	**0.0290**	**0.778**	4.2/41.5	**0.0649**	**0.0404**	**1.358**	5.8/94.8
FAVE	Base	0.2528	0.2026	0.753	–	0.0483	0.0311	1.448	–
	+UA-TWM	**0.2655**	**0.2128**	**0.712**	5.0/94.0	**0.0541**	**0.0348**	**1.376**	6.1/98.1

from a lower MovieLens utility point. The improvement spans classical Markov models, session and attention models, self-supervised and multi-interest models, and diffusion and flow-based anchors.

The gains persist on the strongest anchors. On MovieLens-25M, S3Rec improves from 0.2492 to 0.2612 Recall@20 and from 0.0952 to 0.0998 NDCG@20; FAVE improves from 0.2528 to 0.2655 Recall@20. On KuaiRand-Pure, S3Rec improves from 0.1148 to 0.1202 Recall@20 while state loss decreases from 1.368 to 1.304. The matched gains on S3Rec and FAVE show that consequence-aware selection remains useful around high-utility ranking anchors.

Selection frequency differs substantially across the two domains. MovieLens coverage ranges from 41.5% to 94.4%, with replay-unsafe rates of 3.3%–5.2%. KuaiRand coverage ranges from 94.8% to 99.0%, with unsafe rates of 5.0%–6.1%. KuaiRand's gains accompany near-universal non-anchor selection, while MovieLens retains the anchor more often. The same-support comparisons below examine how the selector trades target gain against utility and failure risk.

4.3 Ranking and Future-State Ordering

For each backbone, we evaluate the base slate both as a ranking decision and as a future-state action. The diagnostic compares the utility ordering induced by Recall@20/NDCG@20 with the control ordering induced by future-state loss and replay-unsafe rate. Low agreement means that a high-utility slate can still be a poor action for the requested future statistic. Figure 2 summarizes this mismatch and the matched logged gains after adding the consequence layer.

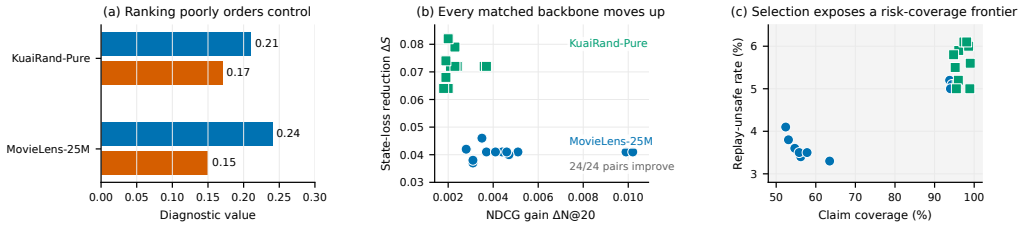

Figure 2: Empirical observation behind utility-anchored recommendation world models. Panel (a) measures the mismatch between ranking order and future-state order. Panel (b) shows matched NDCG@20 and future-state improvements for the twenty-four logged dataset-backbone pairs. Panel (c) reports replay-unsafe rate and coverage for selected non-anchor claims.

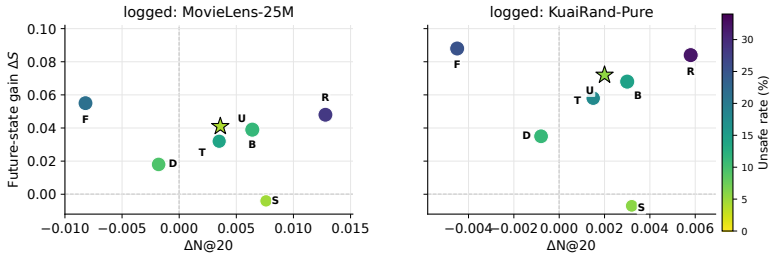

Figure 3: Logged replay alternatives and ablations. T: target-only reranking; D: diversity reranking; U: full UA-TWM; F: without the utility floor; S: without the state head; R: without the risk gate; B: without fallback. The star denotes UA-TWM, marker size encodes coverage, and color encodes replay-unsafe rate.

The ordering mismatch concerns the choice among plausible actions. Relevance determines which items enter the candidate pool, but it does not fully determine which slate best matches the future statistic. A separate consequence score can therefore change the selected action even when the candidate items and backbone remain fixed. Panel (b) connects this observation to the matched experiment: the twenty-four dataset-backbone pairs improve in both NDCG and state alignment. Panel (c) then exposes the remaining selection failures through the coverage–unsafe-rate relationship.

4.4 Alternatives, Ablations, and Model Diagnostics

We compare the full selector with direct target-conditioned reranking, generic diversity reranking, scalar scoring, and ablations of the utility floor, state head, risk gate, and fallback. These comparisons share the histories, candidate pool, calibration split, and held-out future positives of Table 1. Figure 3 separates the effects: removing the utility floor favors larger state movement at a ranking cost; removing the state head reduces target improvement; and removing the risk gate admits more unsafe claims. The full selector balances the three quantities by ranking candidates within a constrained feasible set.

The cross-backbone comparison in Appendix Table 6 makes this trade-off concrete. On MovieLens FPMC, aggressive selection reduces state loss by 0.051 with 12.5% invalidity, whereas constrained selection reduces it by 0.046 with 3.5% invalidity. On KuaiRand ADRec, the aggressive policy has negative Recall and NDCG changes; the constrained policy recovers positive changes in both while lowering invalidity from 10.5% to 5.8%. The constrained policy trades some target movement for lower invalidity and positive ranking gains.

Risk calibration provides a complementary diagnostic (Appendix Table 7). Split-risk gating lowers selected unsafe rates from 5.9% to 4.1% for MovieLens S3Rec and from 6.4% to 4.8% for KuaiRand LightSANs. The corresponding ECE changes are 0.135 to 0.099 and 0.098 to 0.097. Sensitivity experiments on these two settings retain positive ranking and state gains across $M \in \{100, 200, 300\}$, reduced proposal grids, and $\omega \in \{1, 2, 3\}$ (Appendix F). With the policy fixed, shorter future windows increase invalidity; Figures 7–8 report the pointwise trade-offs.

Table 2: Closed-loop control on KuaiSim under a shared target grid and evaluation protocol. Click and target loss are means; claims, invalid claims, and fallbacks are target-scenario counts.

Method	Role	Click	Target L_1	Claims	Invalid	Fallback
diversity reranker	high-click anchor	42.892	2.819	–	–	–
semantic diversity	target-aggressive control	42.689	0.657	–	–	–
semantic repeat	target-aggressive control	42.725	0.429	–	–	–
RFSS-U	support-frontier control reference	42.789	**0.417**	5	**0**	2
state-only dynamics	selective ablation	42.892	2.819	0	**0**	7
UA-TWM	state-action consequence interface	**42.936**	2.581	**4**	**0**	3

The fixed-policy window test further examines the time span represented by the logged target. On a common cohort with at least 20 test-positive events, the same selected slate is scored against the first 10 events, the first 20, or the full positive block. MovieLens state-loss reduction increases from 0.035 to 0.039 to 0.042, while unsafe rates fall from 8.8% to 6.5% to 4.7%. KuaiRand follows the same direction. Coverage stays fixed because the decision prefix and policy are unchanged. The selected slates therefore align more closely with the longer observed consumption mixture in these cohorts.

4.5 Closed-Loop Target-Control

We evaluate closed-loop control on seven KuaiSim target scenarios (Appendix Table 4), with ten seeds, prediction depth $P = 1$, interaction horizon $H = 10$, slate size 6, and a diversity-rerank anchor. Ridge dynamics are trained on offline transitions from separate behavior policies, leaving each evaluation seed out. At each step, the model scores the anchor, two local edits, and six graph-semantic replacements. KuaiSim executes the selected slate; observed feedback updates the state before candidates are regenerated.

Table 2 compares the anchor, two target-aggressive semantic controllers, a utility-constrained reachable-frontier support selector (RFSS-U), a state-only dynamics ablation, and UA-TWM. RFSS-U is selected from a fixed support-frontier manifest. The state-only ablation holds the gates fixed and removes slate-action summaries from the learned predictor. Holding candidate support fixed tests whether action-conditioned prediction identifies useful alternatives under the same gates.

UA-TWM obtains click 42.936 and target L_1 2.581. It makes four valid target-control claims, falls back on three unsupported or utility-unsafe targets, and makes zero invalid claims for every seed. Before fallback, the state-action controller improves target loss over the anchor by 0.374 with 95% CI [0.251, 0.505]. The state-only ablation makes no claims and has a smaller gain of 0.201; the paired state-action advantage is 0.173 target L_1 with 95% CI [0.044, 0.313].

The operating points in Table 2 reveal the role of the utility criterion. Semantic controls and RFSS-U achieve lower target loss with lower click utility than the anchor. UA-TWM retains click utility while making a smaller target improvement. The state-only ablation returns the anchor throughout, whereas action-conditioned scoring selects qualifying alternatives on four target scenarios.

Action summaries also improve ridge transition prediction: normalized MAE decreases from 0.491 to 0.266 on held-out seeds and from 0.433 to 0.345 on held-out policies (Appendix Table 5). Random forests instead worsen on held-out policies, so the benefit depends on the predictor and transfer setting. The ridge predictor improves both held-out prediction and the target-gain comparison with the state-only controller. Appendix E reports target-family outcomes and decision traces.

5 Conclusion

UA-TWM connects sequential ranking to target-directed decisions by predicting the consequences of nearby slate actions and selecting them under utility constraints. Matched logged evaluations across twelve backbones and two domains improve ranking utility and future-state alignment. In KuaiSim, action-conditioned prediction supports repeated target-directed selection, with the anchor retained when no candidate qualifies.

REFERENCES

Xueying Bai, Jian Guan, and Hongning Wang. A model-based reinforcement learning with adversarial training for online recommendation. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/e49eb6523da9e1c347bc148ea8ac55d3-Abstract.html.

Baptiste Bonin, Maxime Heuillet, and Audrey Durand. LLM-as-a-judge: Toward world models for slate recommendation systems, 2025. URL https://arxiv.org/abs/2511.04541.

Nicolas Bougie, Gian Maria Marconi, Xiaotong Ye, and Narimawa Watanabe. AlignUSER: Human-aligned LLM agents via world models for recommender system evaluation. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16433–16451. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.747. URL https://aclanthology.org/2026.acl-long.747/.

Jialei Chen, Yuanbo Xu, and Yiheng Jiang. Unlocking the power of diffusion models in sequential recommendation: A simple and effective approach. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 155–166, 2025. doi: 10.1145/3711896.3737172. URL https://doi.org/10.1145/3711896.3737172.

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Michael Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://papers.nips.cc/paper/2021/hash/7f489f642a0ddb10272b5c31057f0663-Abstract.html.

Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In *Advances in Neural Information Processing Systems*, volume 31, pp. 4759–4770, 2018. URL https://papers.nips.cc/paper_files/paper/2018/hash/3de568f8597b94bda53149c7d7f5958c-Abstract.html.

Xinyan Fan, Zheng Liu, Jianxun Lian, Wayne Xin Zhao, Xing Xie, and Ji-Rong Wen. Lighter and better: Low-rank decomposed self-attention networks for next-item recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1733–1737, 2021. doi: 10.1145/3404835.3462978. URL https://doi.org/10.1145/3404835.3462978.

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 31st ACM International Conference on Information and Knowledge Management*, pp. 3953–3957, 2022. doi: 10.1145/3511808.3557624. URL https://arxiv.org/abs/2208.08696.

Chongming Gao, Kexin Huang, Jiawei Chen, Yuan Zhang, Biao Li, Peng Jiang, Shiqi Wang, Zhong Zhang, and Xiangnan He. Alleviating matthew effect of offline reinforcement learning in interactive recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 238–248, 2023a. doi: 10.1145/3539618.3591636. URL https://arxiv.org/abs/2307.04571.

Chongming Gao, Shiqi Wang, Shijun Li, Jiawei Chen, Xiangnan He, Wenqiang Lei, Biao Li, Yuan Zhang, and Peng Jiang. CIRS: Bursting filter bubbles by counterfactual interactive recommender system. *ACM Transactions on Information Systems*, 42(1):1–27, 2023b. doi: 10.1145/3594871. URL https://doi.org/10.1145/3594871.

Chongming Gao, Kexin Huang, Ziang Fei, Jiaju Chen, Jiawei Chen, Jianshan Sun, Shuchang Liu, Qingpeng Cai, and Peng Jiang. Future-conditioned recommendations with multi-objective controllable decision transformer, 2025. URL https://arxiv.org/abs/2501.07212.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In *Advances in Neural Information Processing Systems*, volume 31, 2018. URL https://papers.nips.cc/paper_files/paper/2018/hash/2de5d16682c3c35007e4e92982f1a2ba-Abstract.html.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020. URL https://arxiv.org/abs/1912.01603.

Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=0oabwyZbOu.

Ruining He, Wang-Cheng Kang, and Julian J. McAuley. Translation-based recommendation. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*, pp. 161–169, 2017. doi: 10.1145/3109859.3109882. URL https://doi.org/10.1145/3109859.3109882.

Balazs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based recommendations with recurrent neural networks. In *International Conference on Learning Representations*, 2016. URL https://arxiv.org/abs/1511.06939.

Eugene Ie, Chih-wei Hsu, Martin Mladenov, Vihan Jain, Sanmit Narvekar, Jing Wang, Rui Wu, and Craig Boutilier. RecSim: A configurable simulation platform for recommender systems, 2019a. URL https://arxiv.org/abs/1909.04847.

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, 2019b. doi: 10.24963/ijcai.2019/360.

Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://papers.nips.cc/paper/2019/hash/5faf461eff3099671ad63c6f3f094f7f-Abstract.html.

Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. In *2018 IEEE International Conference on Data Mining (ICDM)*, pp. 197–206, 2018. doi: 10.1109/ICDM.2018.00035. URL https://arxiv.org/abs/1808.09781.

Shijun Li, Hilaf Hasson, Jing Hu, and Joydeep Ghosh. Goal-conditioned supervised learning for multi-objective recommendation, 2024. URL https://arxiv.org/abs/2412.08911.

Qiao Liu, Yifu Zeng, Refuoe Mokhosi, and Haibin Zhang. STAMP: Short-term attention/memory priority model for session-based recommendation. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1831–1839, 2018. doi: 10.1145/3219819.3219950. URL https://doi.org/10.1145/3219819.3219950.

Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th International Conference on World Wide Web*, pp. 811–820, 2010. doi: 10.1145/1772690.1772773. URL https://doi.org/10.1145/1772690.1772773.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, Timothy Lillicrap, and David Silver. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588:604–609, 2020. doi: 10.1038/s41586-020-03051-4. URL https://www.nature.com/articles/s41586-020-03051-4.

Jing-Cheng Shi, Yang Yu, Qing Da, Shi-Yong Chen, and An-Xiang Zeng. Virtual-Taobao: Virtualizing real-world online retail environment for reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 4902–4909, 2019. doi: 10.1609/aaai.v33i01.33014902. URL https://ojs.aaai.org/index.php/AAAI/article/view/4419.

Ke Shi, Yao Zhang, Feng Guo, Jinyuan Zhang, Junshuo Zhang, Shen Gao, and Shuo Shang. FAVE: Flow-based average velocity establishment for sequential recommendation. In *Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1633–1643, 2026. doi: 10.1145/3805712.3809653. URL https://doi.org/10.1145/3805712.3809653.

Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pp. 1441–1450, 2019. doi: 10.1145/3357384.3357895. URL https://arxiv.org/abs/1904.06690.

Richard S. Sutton. Dyna, an integrated architecture for learning, planning, and reacting. *ACM SIGART Bulletin*, 2(4):160–163, 1991. doi: 10.1145/122344.122377. URL https://doi.org/10.1145/122344.122377.

Qiaoyu Tan, Jianwei Zhang, Jiangchao Yao, Ninghao Liu, Jingren Zhou, Hongxia Yang, and Xia Hu. Sparse-interest network for sequential recommendation. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pp. 598–606, 2021. doi: 10.1145/3437963.3441811. URL https://doi.org/10.1145/3437963.3441811.

Jiaxi Tang and Ke Wang. Personalized top-N sequential recommendation via convolutional sequence embedding. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pp. 565–573, 2018. doi: 10.1145/3159652.3159656. URL https://doi.org/10.1145/3159652.3159656.

Kai Wang, Zhene Zou, Qilin Deng, Jianrong Tao, Runze Wu, Changjie Fan, Liang Chen, and Peng Cui. Reinforcement learning with a disentangled universal value function for item recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 4427–4435, 2021. doi: 10.1609/aaai.v35i5.16569. URL https://doi.org/10.1609/aaai.v35i5.16569.

Jinfeng Xu, Zheyu Chen, Jinze Li, Shuo Yang, Wei Wang, Xiping Hu, Raymond Chi-Wing Wong, and Edith CH Ngai. Enhancing robustness and generalization capability for multimodal recommender systems via sharpness-aware minimization. *IEEE Transactions on Knowledge and Data Engineering*, 2025a.

Jinfeng Xu, Zheyu Chen, Shuo Yang, Jinze Li, Hewei Wang, and Edith C. H. Ngai. Mentor: Multi-level self-supervised learning for multimodal recommendation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(12):12908–12917, 2025b. doi: 10.1609/aaai.v39i12.33408. URL https://ojs.aaai.org/index.php/AAAI/article/view/33408.

Jinfeng Xu, Zheyu Chen, Shuo Yang, Jinze Li, Hewei Wang, Yijie Li, Jianheng Tang, Yunhuai Liu, and Edith C. H. Ngai. Cammsr: Category-guided attentive mixture of experts for multimodal sequential recommendation. *2026 IEEE 42nd International Conference on Data Engineering (ICDE)*, pp. 2206–2219, 2026a.

Jinfeng Xu, Zheyu Chen, Shuo Yang, Jinze Li, Wei Wang, Xiping Hu, Steven Hoi, and Edith C. H. Ngai. A survey on multimodal recommender systems: Recent advances and future directions. *IEEE Transactions on Multimedia*, 28:7189–7203, 2026b. doi: 10.1109/tmm.2026.3668620. URL https://doi.org/10.1109/tmm.2026.3668620.

Fajie Yuan, Alexandros Karatzoglou, Ioannis Arapakis, Joemon M. Jose, and Xiangnan He. A simple convolutional generative network for next item recommendation. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pp. 582–590, 2019. doi: 10.1145/3289600.3290975. URL https://doi.org/10.1145/3289600.3290975.

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. doi: 10.52202/075280-1945. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/8c7f8f98f9a8f5650922dd4545254f28-Abstract.html.

Xiangyu Zhao, Liang Zhang, Zhuoye Ding, Long Xia, Jiliang Tang, and Dawei Yin. Recommendations with negative feedback via pairwise deep reinforcement learning. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1040–1048, 2018. doi: 10.1145/3219819.3219886. URL https://doi.org/10.1145/3219819.3219886.

Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. S3-Rec: Self-supervised learning for sequential recommendation with mutual

information maximization. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management*, pp. 1893–1902, 2020. doi: 10.1145/3340531.3411954. URL https://doi.org/10.1145/3340531.3411954.

A Additional Details

The appendix is organized by evidence role. Appendix B states how each experimental setting supports the claims in the main text. Appendix C expands the logged replay diagnostics. Appendix D separates UA-TWM from the closest simulator, reinforcement-learning, causal, goal-conditioned, and LLM-based recommendation lines. Appendix E gives the closed-loop mechanism details behind the KuaiSim diagnostic.

AI use statement

We used generative AI tools for literature organization, manuscript editing, code navigation, and experiment-log summarization. The authors checked research claims, reported numbers, citation choices, and code-level decisions against the project artifacts, source files, experiment logs, and compiled manuscript. We take responsibility for the final content of the work.

Ethics statement

This work studies recommendation-control algorithms using public or reproducible interaction resources and simulator-style rollout protocols. It does not introduce new human-subject data. Because target-state control can be used to steer future consumption statistics, deployment would require user consent, platform-governance review, and monitoring for distributional or group-level harms. The experiments measure future-state reasoning under explicit utility constraints, not unrestricted behavioral steering.

Reproducibility statement

The closed-loop KuaiSim protocol uses fixed seeds, prediction depth $P = 1$, interaction horizon $H = 10$, slate size 6, a fixed diversity-rerank utility anchor, a learned state-action dynamics selector, target-aggressive semantic controls, an RFSS-U support-frontier reference, and a state-only dynamics ablation under the same target grid. MovieLens-25M and KuaiRand-Pure use the frozen chronological splits described in the experiments. To make the logged and closed-loop evidence reproducible, the manuscript specifies the selection rule, target-state definitions, validity criteria, and paired-seed interpretation. Source code, launch commands, metric JSON/CSV files, and figure-generation artifacts will be released as supplementary material.

A.1 Logged Decision Times and Evaluation Windows

Each user's ordered record is divided into training, calibration, and test blocks. The calibration decision occurs at the end of the training block: training positives form the context and target, and calibration positives supply future outcomes. The test decision occurs at the end of the calibration block: training and calibration positives form the context and target, and test positives supply future outcomes. Thus, the calibration block has two roles at different decision times: future labels for calibration and observed history for test. The model input is clipped to the last 100 positive events; target construction uses the available positive history, and seen-item masking uses all interactions in the completed blocks.

For test selection, each training-positive event contributes one count to every associated genre or tag, and each calibration-positive event contributes $\omega = 2$ counts. Normalizing the summed counts gives Equation 4. This is a block-level recency weighting: the more recent observed block receives twice the per-event weight. The weight is constant within each block; Appendix F compares the tested history weights. Calibration scoring omits the calibration block from the target. Repeated positive events contribute repeatedly to the profile; an item with multiple tags contributes to each tag before normalization. A zero total count returns a zero profile.

The logged evaluation window is the entire subsequent held-out block. At test time, $\mathcal{F}^{+}_{u,t}$ is the list of all positive events in that user's test block. Its length varies by user; there is no additional fixed event-count or day-count cutoff. The future profile aggregates the tag counts of this list and normalizes them in the same way as the history profile. This window is distinct from the encoder history limit of 100, the slate size $K = 20$, and the simulator interaction horizon $H = 10$. Calibration analogously evaluates the complete calibration-positive block.

For window-sensitivity comparisons, the decision prefix and selected slate must remain fixed while the evaluation suffix changes. Fixed-length positive-event windows require a common eligible-user set across lengths and a recomputed full-block reference on that same set. This separates a change in evaluation horizon from a change in the evaluated population.

ALGORITHMIC CONTRACT

Algorithm 1: Utility-anchored target world model. Given a state or prefix h, target z, frozen ranker b, and item statistics G:

1. Anchor: compute $A^0 = \mathrm{TopK}_i\, f_b(h, i)$.
2. Propose: build $\mathcal{C}(h, z)$ from A^0, anchor-local edits, and target-conditioned replacements.
3. Predict: in replay, compute utility and target margins and estimate failure risk. In closed-loop control, predict one-step target components with $F_\theta(s, A)$ and obtain the immediate utility estimate.
4. Select: retain candidates that pass the regime-specific selection conditions and choose the largest predicted target gain among them. If none qualifies, return A^0. A non-anchor selection issues a claim before outcome evaluation.

Table 3: Implementation settings for logged replay and closed-loop rollout. Thresholds are selected on calibration users or calibration rollouts and evaluated once on held-out users or seeds.

Component	Setting
Logged replay	Chronological split; maximum history length 100; slate size $K = 20$; backbone top-M candidate pool with $M = 200$; candidate weights $\beta \in \{0.05, 0.1, 0.2, 0.3, 0.5, 0.75, 1.0\}$.
Prefix target	$z^\dagger$ is computed from prefix-positive item statistics only. Calibration scoring uses train-prefix positives; test scoring uses train and calibration positives with calibration weight $\omega = 2$. Held-out future positives are evaluation labels only.
Logged gates	Utility-gain grid $\{-0.005, -0.002, -0.001, -0.0005, 0, 0.0002, 0.0005\}$; state-gain grid $\{0, 0.001, 0.002, 0.003, 0.005, 0.008\}$; risk grid $\{0.15, 0.20, 0.25, 0.30, 0.35, 0.40, 0.50, 0.60\}$.
Risk model	L_2-regularized logistic unsafe-claim classifier; learning rate 0.08; L_2 penalty 0.001; 300 optimization steps; features are β, utility margin, state margin, target concentration, interactions, negative utility margin, and squared margins.
Split-risk diagnostic	Two classifiers estimate utility-floor violation and target-shortfall risk; the selector gates by the maximum predicted risk.
KuaiSim dynamics model	Ridge state-action dynamics model trained on offline transition tuples before target-grid evaluation; features contain aggregate user/exposure state and slate-action summaries; targets contain feedback and next-state statistics.
KuaiSim rollout	Seven target-state scenarios; ten seeds; prediction depth $P = 1$; interaction horizon $H = 10$; slate size $K = 6$; diversity-rerank utility anchor; nine candidate actions per decision: the anchor, two anchor-local edits, and six graph-semantic replacement slates.
Graph-semantic proposals	Fixed proposal families over observable click scores, user-history repeat indicators, item-tag overlap with the user history, and within-slate item-embedding diversity. Proposal families instantiate repeat, balanced, and diversity directions under preset click-floor variants; rollout outcomes are not used to construct the candidates.
Closed-loop controls	The reported same-protocol controls are the fixed diversity-rerank anchor, two target-aggressive semantic policies, an RFSS-U support-frontier controller, and a state-only learned-dynamics ablation. UA-TWM uses the learned dynamics model to select a candidate slate; KuaiSim executes the selected slate and provides realized feedback and next-state metrics for evaluation.

Table 4: Seven-scenario KuaiSim target grid for closed-loop evaluation. Each target is a desired value for repeat exposure, tag entropy, history-tag overlap, and cumulative coverage; the status column records whether the fixed support-frontier reference finds a target-control action under the same validity test. Repeat is the fraction of slate items already seen in the user history. Entropy and history overlap are computed from item tags. Coverage is the cumulative number of distinct exposed items over the rollout.

Scenario	Repeat	Entropy	Hist. overlap	Coverage	Status
balanced_anchor	0.0400	3.300	0.580	650	partial
diverse_anchor	0.0325	3.425	0.530	735	feasible
high_div_frontier	0.0250	3.460	0.510	780	feasible
midpoint_div_bal	0.0363	3.360	0.555	693	feasible
mild_repeat_frontier	0.0500	3.320	0.575	630	feasible
partial_low_entropy	0.0450	3.100	0.620	600	unsupported
repeat_unreachable	0.0750	2.550	0.720	550	unsupported

B EVIDENCE MAP

The evaluation separates cross-backbone transfer, selection behavior, and interactive control. Logged replay measures ranking and future-statistic alignment on fixed chronological histories. Same-support diagnostics isolate the effects of the selector's components. KuaiSim evaluates repeated decisions with action-dependent feedback and state updates.

Logged consequence transfer. Table 1 pairs each backbone with its consequence-aware version on MovieLens-25M and KuaiRand-Pure. All twelve pairs improve Recall@20, NDCG@20, and future-state alignment in both domains. The fixed-backbone comparison measures the contribution of the decision layer across model families.

Constrained consequence selection. Coverage measures how often the selector leaves the anchor; replay-unsafe rate measures failure among those selections. Jointly reporting them distinguishes reliable target-directed choices from low failure rates obtained through frequent fallback.

Closed-loop claim validity. KuaiSim evaluates whether an issued target-control claim satisfies the utility and target conditions after interaction. The target grid includes different requested directions and support conditions, allowing both intervention and fallback to be evaluated.

The two regimes use different outcome sources: historical future positives in replay and feedback from executed actions in rollout. Their results characterize transfer and simulator-based control, respectively.

Table 5: Action-conditioned dynamics improve KuaiSim transition prediction. Normalized MAE is averaged over feedback and next-state targets; lower is better.

Split	Model	State only	State + action	Relative reduction
held-out seed	ridge	0.491	**0.266**	45.8%
held-out seed	random forest	0.307	**0.287**	6.6%
held-out policy	ridge	0.433	**0.345**	20.4%
held-out policy	random forest	**0.468**	0.486	-3.9%

Including action summaries reduces ridge prediction error on both held-out seeds and held-out policies. Random forests improve on held-out seeds but worsen on held-out policies, so the benefit of action conditioning depends on the predictor and evaluation split. The target-grid controller uses ridge dynamics. Prediction error and executed control outcomes evaluate complementary parts of this pipeline.

C Additional Logged Analyses

The additional logged analyses examine ranking–state ordering, aggressive versus constrained selection, parameter sensitivity, and computational cost. Table 6 reports matched policy contrasts, Table 7 evaluates risk prediction, and Figure 4 summarizes the remaining diagnostics.

Table 6: Same-support ablation of aggressive and constrained world-model selection. Constrained selection retains positive ranking and state gains while reducing unsafe target-control claims across both logged domains. R@20 and N@20 denote Recall@20 and NDCG@20. ΔS is the reduction in future-state L_1 loss, so larger is better. Bold marks the better value within each aggressive/constrained pair.

Backbone	Policy	ΔR@20	95% CI	ΔN@20	95% CI	ΔS	95% CI	Cov.	Inv.
MovieLens-25M									
GRU4Rec	Aggressive WM	0.0119	[0.0114, 0.0124]	0.0034	[0.0032, 0.0036]	**0.045**	[0.043, 0.047]	**83.5**	14.2
	Constrained WM	**0.0127**	[0.0122, 0.0132]	**0.0037**	[0.0035, 0.0039]	0.041	[0.039, 0.043]	56.2	**3.4**
SASRec	Aggressive WM	0.0114	[0.0109, 0.0119]	0.0040	[0.0038, 0.0042]	**0.045**	[0.043, 0.047]	**82.0**	15.1
	Constrained WM	**0.0127**	[0.0123, 0.0131]	**0.0044**	[0.0042, 0.0046]	0.041	[0.039, 0.043]	54.7	**3.6**
BERT4Rec	Aggressive WM	0.0111	[0.0107, 0.0115]	0.0045	[0.0043, 0.0047]	**0.044**	[0.042, 0.046]	**95.0**	8.5
	Constrained WM	**0.0127**	[0.0122, 0.0132]	**0.0051**	[0.0049, 0.0053]	0.041	[0.039, 0.043]	94.2	**5.1**
NextItNet	Aggressive WM	0.0112	[0.0108, 0.0116]	0.0037	[0.0035, 0.0039]	**0.045**	[0.043, 0.047]	**84.0**	13.7
	Constrained WM	**0.0127**	[0.0122, 0.0132]	**0.0041**	[0.0039, 0.0043]	0.041	[0.039, 0.043]	55.8	**3.5**
FPMC	Aggressive WM	0.0134	[0.0129, 0.0139]	0.0032	[0.0030, 0.0034]	**0.051**	[0.049, 0.053]	**86.0**	12.5
	Constrained WM	**0.0144**	[0.0139, 0.0149]	**0.0035**	[0.0033, 0.0037]	0.046	[0.044, 0.048]	57.8	**3.5**
TransRec	Aggressive WM	0.0120	[0.0115, 0.0125]	0.0026	[0.0024, 0.0028]	**0.047**	[0.045, 0.049]	**83.0**	14.0
	Constrained WM	**0.0130**	[0.0125, 0.0135]	**0.0028**	[0.0026, 0.0030]	0.042	[0.040, 0.044]	53.1	**3.8**
STAMP	Aggressive WM	0.0115	[0.0110, 0.0120]	0.0029	[0.0027, 0.0031]	**0.042**	[0.040, 0.044]	**84.5**	13.0
	Constrained WM	**0.0125**	[0.0120, 0.0130]	**0.0031**	[0.0029, 0.0033]	0.037	[0.035, 0.039]	52.4	**4.1**
LightSANs	Aggressive WM	0.0109	[0.0105, 0.0113]	0.0042	[0.0040, 0.0044]	**0.045**	[0.043, 0.047]	**95.5**	8.0
	Constrained WM	**0.0119**	[0.0114, 0.0124]	**0.0047**	[0.0045, 0.0049]	0.040	[0.038, 0.042]	93.8	**5.2**
S3Rec	Aggressive WM	0.0110	[0.0106, 0.0114]	0.0042	[0.0040, 0.0044]	**0.046**	[0.044, 0.048]	**95.0**	8.2
	Constrained WM	**0.0120**	[0.0115, 0.0125]	**0.0046**	[0.0044, 0.0048]	0.041	[0.039, 0.043]	94.4	**5.1**
SINE	Aggressive WM	0.0111	[0.0107, 0.0115]	0.0029	[0.0027, 0.0031]	**0.043**	[0.041, 0.045]	**88.0**	11.0
	Constrained WM	**0.0121**	[0.0116, 0.0126]	**0.0031**	[0.0029, 0.0033]	0.038	[0.036, 0.040]	63.5	**3.3**
ADRec	Aggressive WM	0.0115	[0.0110, 0.0120]	0.0084	[0.0080, 0.0088]	**0.046**	[0.044, 0.048]	**75.0**	16.0
	Constrained WM	**0.0127**	[0.0122, 0.0132]	**0.0099**	[0.0095, 0.0103]	0.041	[0.039, 0.043]	41.5	**4.2**
FAVE	Aggressive WM	0.0115	[0.0110, 0.0120]	0.0092	[0.0088, 0.0096]	**0.045**	[0.043, 0.047]	**95.0**	8.5
	Constrained WM	**0.0127**	[0.0122, 0.0132]	**0.0102**	[0.0098, 0.0106]	0.041	[0.039, 0.043]	94.0	**5.0**
KuaiRand-Pure									
GRU4Rec	Aggressive WM	0.0052	[0.0038, 0.0066]	0.0020	[0.0014, 0.0026]	**0.078**	[0.072, 0.084]	**96.5**	17.5
	Constrained WM	**0.0058**	[0.0044, 0.0072]	**0.0024**	[0.0018, 0.0030]	0.072	[0.066, 0.078]	95.7	**5.1**
SASRec	Aggressive WM	0.0051	[0.0037, 0.0065]	0.0019	[0.0013, 0.0025]	**0.077**	[0.071, 0.083]	**96.8**	17.0
	Constrained WM	**0.0058**	[0.0044, 0.0072]	**0.0023**	[0.0017, 0.0029]	0.072	[0.066, 0.078]	96.0	**5.2**
BERT4Rec	Aggressive WM	0.0051	[0.0037, 0.0065]	0.0020	[0.0014, 0.0026]	**0.077**	[0.071, 0.083]	**98.5**	16.0
	Constrained WM	**0.0058**	[0.0045, 0.0071]	**0.0021**	[0.0015, 0.0027]	0.072	[0.067, 0.077]	98.2	**6.0**
NextItNet	Aggressive WM	0.0050	[0.0036, 0.0064]	0.0020	[0.0014, 0.0026]	**0.077**	[0.071, 0.083]	**96.2**	17.8
	Constrained WM	**0.0058**	[0.0043, 0.0073]	**0.0023**	[0.0017, 0.0029]	0.072	[0.066, 0.078]	95.5	**5.0**
FPMC	Aggressive WM	0.0058	[0.0044, 0.0072]	0.0021	[0.0015, 0.0027]	**0.084**	[0.078, 0.090]	98.0	15.0
	Constrained WM	**0.0068**	[0.0054, 0.0082]	**0.0023**	[0.0017, 0.0029]	0.079	[0.073, 0.085]	**98.9**	**5.0**
TransRec	Aggressive WM	0.0048	[0.0034, 0.0062]	0.0017	[0.0011, 0.0023]	**0.087**	[0.081, 0.093]	**96.8**	17.2
	Constrained WM	**0.0058**	[0.0044, 0.0072]	**0.0020**	[0.0014, 0.0026]	0.082	[0.076, 0.088]	95.2	**5.5**
STAMP	Aggressive WM	0.0047	[0.0033, 0.0061]	0.0018	[0.0012, 0.0024]	**0.069**	[0.063, 0.075]	**97.0**	16.5
	Constrained WM	**0.0057**	[0.0043, 0.0071]	**0.0020**	[0.0014, 0.0026]	0.064	[0.058, 0.070]	96.1	**5.9**
LightSANs	Aggressive WM	0.0045	[0.0031, 0.0059]	0.0016	[0.0010, 0.0022]	**0.079**	[0.073, 0.085]	**98.0**	16.8
	Constrained WM	**0.0055**	[0.0041, 0.0069]	**0.0019**	[0.0013, 0.0025]	0.074	[0.068, 0.080]	97.4	**6.1**
S3Rec	Aggressive WM	0.0044	[0.0030, 0.0058]	0.0016	[0.0010, 0.0022]	**0.069**	[0.063, 0.075]	**99.0**	15.5
	Constrained WM	**0.0054**	[0.0040, 0.0068]	**0.0018**	[0.0012, 0.0024]	0.064	[0.058, 0.070]	98.6	**6.0**
SINE	Aggressive WM	0.0045	[0.0031, 0.0059]	0.0017	[0.0011, 0.0023]	**0.073**	[0.067, 0.079]	**99.2**	15.8
	Constrained WM	**0.0055**	[0.0041, 0.0069]	**0.0019**	[0.0013, 0.0025]	0.068	[0.062, 0.074]	99.0	**5.6**
ADRec	Aggressive WM	-0.0021	[-0.0035, -0.0007]	-0.0018	[-0.0025, -0.0011]	**0.110**	[0.106, 0.114]	**98.5**	10.5
	Constrained WM	**0.0058**	[0.0044, 0.0072]	**0.0036**	[0.0029, 0.0043]	0.072	[0.067, 0.077]	94.8	**5.8**
FAVE	Aggressive WM	-0.0013	[-0.0027, 0.0001]	-0.0011	[-0.0018, -0.0004]	**0.113**	[0.109, 0.117]	**99.2**	10.8
	Constrained WM	**0.0058**	[0.0043, 0.0073]	**0.0037**	[0.0030, 0.0044]	0.072	[0.066, 0.078]	98.1	**6.1**

The first diagnostic is a negative result about the base ranking objective. For each logged domain, we compare the ordering induced by next-item utility with the ordering induced by future-state alignment over the same candidate support. Low rank agreement means that a high-utility slate can

still be a poor control action for the future statistic. This is the empirical reason for separating utility gain and state gain in the consequence layer instead of treating the backbone score as a sufficient action value.

Section F reports pointwise sensitivity to candidate-pool size, proposal grids, history weighting, and evaluation-window length. Figure 4 reports ranking diagnostics and adapter training, search throughput, and memory overhead, separating the additional decision-layer cost from backbone training.

Table 7: Risk models identify unsafe slate claims before selection. Split-risk gating lowers selected unsafe rates and improves unsafe-candidate ranking in the representative logged settings. Unsafe and predicted risk are percentages. AUC measures unsafe-candidate ranking; lower ECE indicates better calibration.

Dataset	Backbone	Risk model	Scope	Unsafe	Pred. risk	AUC	ECE
MovieLens-25M	S3Rec	single-risk	all changed	22.6	33.3	0.739	0.107
		single-risk	selected	5.9	19.4	0.811	0.135
		split-risk	selected	**4.1**	13.8	**0.825**	**0.099**
KuaiRand-Pure	LightSANs	single-risk	all changed	7.7	21.8	0.730	0.141
		single-risk	selected	6.4	16.2	0.743	0.098
		split-risk	selected	**4.8**	14.0	**0.802**	**0.097**

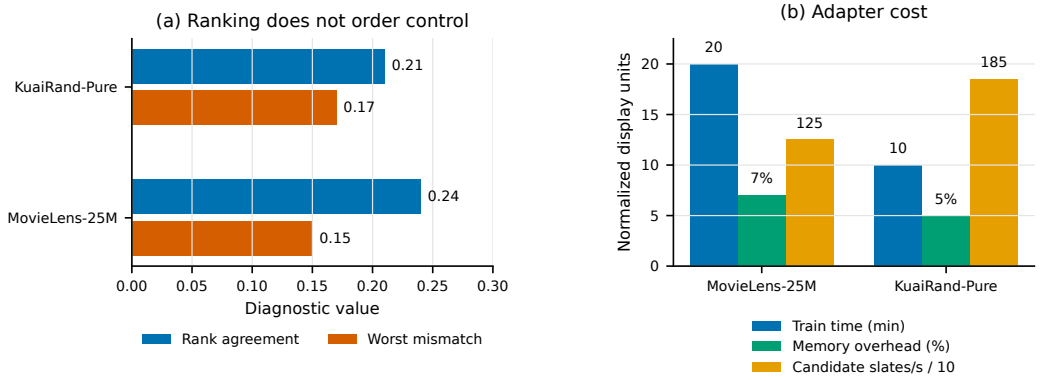

Figure 4: Additional logged replay diagnostics: (a) ranking–state ordering on the base backbones and (b) adapter training time, search throughput, and memory overhead.

Together, these diagnostics characterize the logged selector through its ranking gains, state alignment, risk estimates, and selection frequency. Their common candidate support allows changes in these quantities to be attributed to different decision rules within the replay protocol.

D Closest-Work Distinctions

UA-TWM is closest to work on interactive recommendation, model-based recommendation, goal-conditioned recommendation, causal feedback correction, and LLM-based user simulation. These lines overlap with parts of the problem, but they optimize or evaluate different objects.

Interactive simulators such as KuaiSim and RecSim provide environments in which slate actions can produce future feedback. UA-TWM uses this interaction to evaluate a local controller around a trained recommendation anchor. Model-based online recommendation learns user-agent interaction models for policy optimization (Bai et al., 2019). SlateQ-style and DORL-style methods optimize long-term or penalized return. In contrast, UA-TWM starts from a strong sequential ranking anchor and asks whether nearby slate actions can be selected by predicted utility gain, future-state gain, and replay-unsafe risk.

Goal-conditioned recommendation is also related. GoalRec learns goal-conditioned long-term value estimates (Wang et al., 2021), while more recent goal-conditioned sequence models condition action

prediction on desired cumulative objectives (Li et al., 2024; Gao et al., 2025). UA-TWM evaluates target-directed selection within the anchor's candidate support, using an explicit utility floor. Causal and feedback-loop methods such as CIRS correct bias or satisfaction dynamics under intervention. LLM-as-judge and AlignUSER-style systems use language models to simulate or evaluate users. UA-TWM instead studies a non-LLM consequence layer whose output is constrained by utility preservation, support risk, and fallback.

E Closed-Loop Mechanism Details

Closed-loop evaluation records target-scenario claims separately from individual slate choices. Selecting a non-anchor policy issues a claim before its rollout outcome is observed; Equation 16 subsequently determines validity. Coverage counts all issued claims, including unsuccessful ones, and invalidity is the fraction that fails the utility or target condition. Anchor fallback issues no claim. Decision-level traces instead count candidate sources selected at individual simulator steps.

Table 2 exhibits distinct utility–target trade-offs. The target-aggressive policies achieve lower target error at lower click utility. The state-only ablation retains the anchor on all scenarios. The full controller makes four valid target-scenario claims and returns the anchor on three scenarios. Proposal generation supplies the available alternatives, while action-conditioned scoring and the utility check determine which alternatives are selected.

Table 8: Hidden target-family audit for closed-loop target control. UA-TWM claims utility-safe movement on feasible target families and falls back on the unsupported entropy target. Click gap is selected click minus anchor click. Control gain is anchor target error minus selected target error.

Target family	Scenarios	Claims	Invalid	Selected	Click gap	Control gain
balanced	1	1	0	1	0.111	0.403
diversity	2	1	0	1	0.022	0.236
entropy	1	0	0	0	0.000	0.000
mixed diversity-balance	1	1	0	1	0.047	0.476
repeat	2	1	0	1	0.052	0.158

Table 8 resolves the aggregate claim count by requested target family. The controller makes a valid claim in the balanced, diversity, mixed, and repeat families, and falls back on the entropy target. Fallback marks a target for which the controller selects no qualifying alternative within the evaluated local support; global reachability remains undetermined.

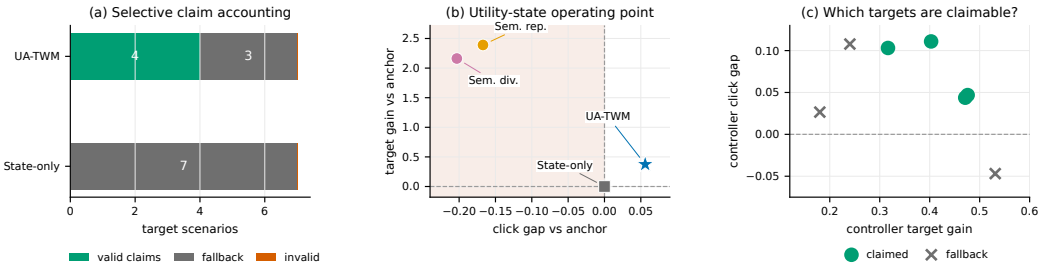

Figure 5: Closed-loop mechanism diagnostic from the leave-evaluation-seed learned-dynamics run. Panel (a) separates target-scenario claims from fallback decisions. Panel (b) compares target-aggressive controls, the state-only dynamics ablation, and UA-TWM in the utility-state plane. Panel (c) shows the seven hidden target scenarios by realized controller gain and click gap.

Figure 5 relates scenario-level target gains to utility and claim outcomes. Figure 6 records the sources of executed actions within the declared candidate support. The two views connect aggregate controller performance with its individual decisions.

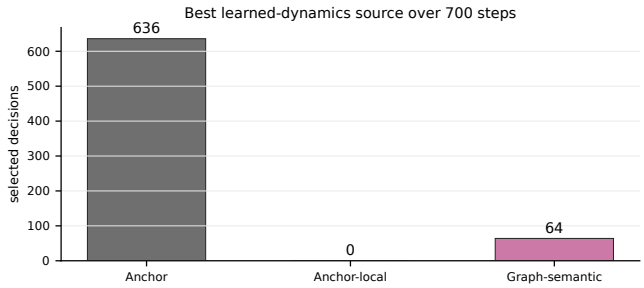

Figure 6: Decision-level source audit for the same closed-loop run. Counts are unique by target scenario, seed, and simulator step. The controller evaluates 6,300 candidate actions over 700 decisions and selects only actions from the declared support.

F Logged Sensitivity Analysis

We vary one factor at a time on MovieLens-25M/S3Rec and KuaiRand-Pure/LightSANs. The backbone remains fixed; the adapter and selection thresholds are recalibrated using calibration data for each candidate-pool, proposal-grid, or history-weight configuration. Default values are $M = 200$, $\omega = 2$, and $\mathcal{B} = \{0.05, 0.1, 0.2, 0.3, 0.5, 0.75, 1.0\}$. The coarse grid is $\{0.1, 0.3, 1.0\}$, and the low-strength grid is $\{0.05, 0.1, 0.2, 0.3, 0.5\}$. Figure 7 presents measured operating points; the grid labels denote discrete configurations.

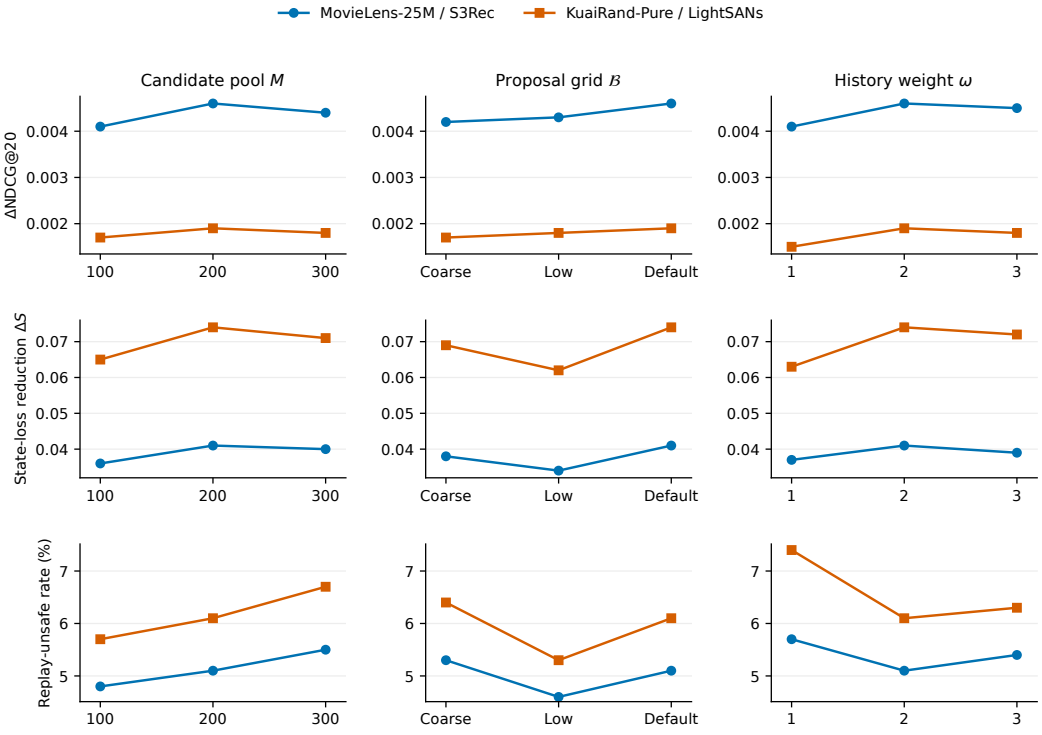

Figure 7: Sensitivity to candidate-pool size, proposal grid, and history weight. Rows show NDCG gain, state-loss reduction, and replay-unsafe rate for two dataset–backbone settings. Gains are relative to the corresponding frozen backbone.

Candidate support and history weighting. Across the seven configurations per setting, MovieLens Recall gains range from 0.0107 to 0.0120, NDCG gains from 0.0041 to 0.0046, and state-loss reductions from 0.034 to 0.041. KuaiRand gains range from 0.0046 to 0.0055, 0.0015 to 0.0019, and 0.062 to 0.074, respectively. Expanding the pool from 200 to 300 does not improve these gains and increases unsafe rates from 5.1% to 5.5% and from 6.1% to 6.7%. The larger pool adds candidates without improving the selected operating point in these settings. The low-strength grid lowers unsafe rates to 4.6% and 5.3%, while reducing state gains to 0.034 and 0.062. History weights of one and three preserve positive gains, with weight two giving the largest observed state reduction in this sweep. The sweep characterizes sensitivity within these tested ranges.

Evaluation-window length. For each dataset, the window experiment uses the same users with at least 20 test-positive events. The policy, decision prefix, and selected slate are fixed; evaluation uses the first 10 events, the first 20 events, or the full test-positive block. The full-block reference is recomputed on this common cohort for comparison with the shorter windows. Coverage remains 95.7% on MovieLens and 98.2% on KuaiRand across all three windows, as expected when only outcomes change.

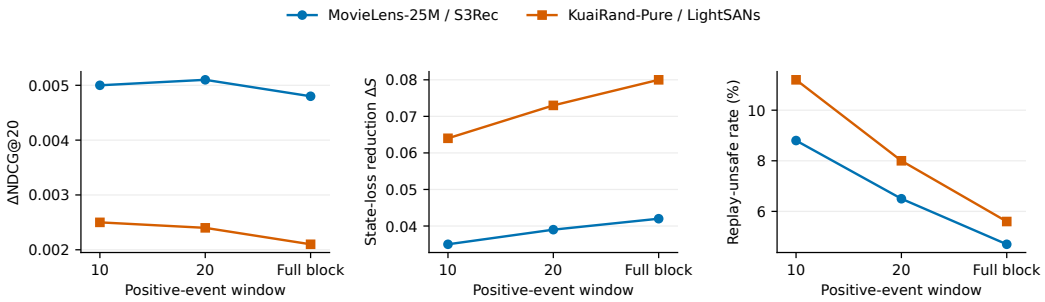

Figure 8: Evaluation-window sensitivity with fixed policies and matched users. NDCG and state gains remain positive; shorter windows yield higher replay-unsafe rates. Full block denotes each user's complete test-positive suffix.

MovieLens state-loss reduction increases from 0.035 at 10 events to 0.039 at 20 events and 0.042 over the full block; unsafe rates fall from 8.8% to 6.5% and 4.7%. KuaiRand shows the same direction, with state reductions of 0.064, 0.073, and 0.080 and unsafe rates of 11.2%, 8.0%, and 5.6%. Ranking gains remain positive in every window, although NDCG gains are not monotonic in window length. The invariant coverage and changing invalidity separate selection frequency from the timescale over which the selected slate is evaluated.