LATENT INTENT, BOUNDED EVIDENCE: REINFORCEMENT LEARNING FOR LANGUAGE-MODEL RECOMMENDERS

Anonymous authors
Paper under double-blind review

ABSTRACT

We introduce PILOT, a method that trains a language-model recommender with reinforcement learning while keeping a language-model critic from inventing reward that the interaction logs cannot support. The difficulty is structural: long-horizon recommendation quality is exactly the quantity that offline logs report least reliably, so any critic accurate enough to be useful is also free enough to be gamed, and policies that chase it drift into slates that score well and serve users poorly. PILOT resolves this by making the critic's licence explicit. A frozen language model infers a discrete latent intent for the current session; a semantic-identifier policy decodes a slate conditioned on that intent; and the reward is composed by allowing a language-model judge to adjust a logged reward ensemble only within the confidence interval that the ensemble's own epistemic uncertainty leaves open. We prove that this composition has a bias envelope proportional to the reward model's uncertainty, and that the resulting policy admits a safe-improvement guarantee against any comparator, whereas an unbounded judge leaves an irreducible value gap no amount of data removes. Across three public recommendation corpora and a calibrated session simulator, PILOT improves both next-item accuracy and long-horizon session utility over sequential, language-model, and reinforcement-learning baselines, and—unlike the baselines it is compared against—its true session utility does not collapse as the policy is pushed further from its reference.

1 INTRODUCTION

Recommender systems are trained on what users did next and evaluated on whether users stayed. Next-item accuracy is cheap to supervise, but it rewards a system for predicting the click the current interface makes most likely, not for composing a session a person is glad to have had. Reinforcement learning is the natural language for the second objective, and a decade of work has formulated recommendation as sequential decision making (Chen et al., 2018; Ie et al., 2019a; Chen et al., 2021). What has held it back is not the formulation but the reward: the long-horizon quantity a recommender should optimise is precisely the one that interaction logs, collected under a deployed policy that never showed most of the catalogue, report least reliably.

Large language models have changed what is available on both sides of this gap. On the policy side they supply world knowledge about items and a generative action space—items emitted as sequences of semantic tokens rather than selected from a flat softmax—that generalises to items the logs never covered (Rajput et al., 2023; Geng et al., 2023; Singh et al., 2023). On the critic side they can be asked whether a slate serves a user's apparent purpose, supplying signal where logged feedback is silent (Lee et al., 2023; Wang et al., 2024). The obvious move is to put the two together: let a language model propose, let a language model judge, and optimise the first against the second.

The obvious move fails in a specific and well-documented way. A judge free to assign reward anywhere is a proxy, and optimising hard against a proxy degrades the quantity the proxy stood in for (Gao et al., 2022c; Bai et al., 2022). A judge asked whether a slate matches a user's interest will happily approve slates that are thematically immaculate and behaviourally inert, and a policy trained against it learns to write plausible-looking recommendations rather than useful ones. The standard remedy—a KL penalty to a reference policy—limits how far the policy moves but says nothing

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.

about *where* the judge should have been trusted. It trades away good exploration together with bad, treating a region the logs cover densely and a region they never touched as equally forbidden.

Our starting observation is that the logs already contain the information needed to decide how much a judge should be believed. A reward model fit to logged feedback is confident where the logging policy explored and uncertain where it did not, and that uncertainty is estimable from an ensemble. This inverts the usual arrangement: rather than treating the language model as the authority and the logs as a regulariser, treat the logs as the evidence and the language model as an interpolator licensed to act only in proportion to what the evidence leaves undetermined. Where behaviour is well observed, the logs decide. Where behaviour was never observed—most of the catalogue, and exactly where generalisation matters—the judge fills in, but never further than the uncertainty it is filling.

We turn this into a method, PILOT, with three parts. A frozen language model infers a *discrete latent intent* for the current session from a small learned codebook, which factorises the policy into a coarse decision about what kind of session this is and a fine decision about which items realise it. A semantic-identifier policy decodes a slate autoregressively under that intent, so exploration happens in a semantically structured space rather than over item indices. And an *evidence-bounded critic* composes the final reward by letting a language-model judge adjust a logged reward ensemble only within a budget set by the ensemble's own epistemic uncertainty, with an additional pessimism term that makes the composite a lower bound on true utility. The policy is then optimised with a two-timescale group-relative scheme that computes advantages at both the intent and the slate level.

The bounded composition is what makes the method analysable. We show that its reward bias is enveloped by the reward model's uncertainty and that the learned policy satisfies a safe-improvement guarantee against any comparator. The same analysis says what goes wrong without the bound: an unconstrained judge contributes a bias term that does not vanish with data, which is the over-optimisation phenomenon stated as a theorem rather than observed as a curve.

Our contributions are as follows.

- We formulate language-model recommendation as a latent-intent-conditioned decision process, in which a discrete intent code mediates between session history and slate generation and carries its own advantage.
- We propose the evidence-bounded reward, which admits language-model judgement in proportion to the epistemic uncertainty of a logged reward ensemble, and prove a bias envelope, a value-gap bound, and a safe-improvement guarantee for the policies it induces.
- We show that removing the bound leaves an irreducible term in the value gap, giving a precise account of why unconstrained language-model critics over-optimise in recommendation.
- We evaluate on three public corpora and a calibrated session simulator, finding gains in both accuracy and long-horizon utility, and a qualitatively different response to policy drift than the baselines show.

2 PROBLEM SETTING

We consider interactive slate recommendation. A session for user u proceeds over turns $t = 1, \dots, T_u$. At turn t the system observes a history $h_t \in \mathcal{H}$ summarising the items shown so far and the feedback they drew, and emits a slate $A_t = (a_t^1, \dots, a_t^K)$ of K distinct items from a catalogue $\mathcal{I}$. The user returns feedback o_t—clicks, dwell, skips, or session exit—which extends the history to h_{t+1}.

This defines a Markov decision process $\mathcal{M} = (\mathcal{H}, \mathcal{I}^K, P, r^\star, \gamma)$ whose kernel P is the user's response process and whose reward $r^\star(h, A)$ is the utility the user actually derives from the slate. The objective is the discounted session return

$$J_{r^\star}(\pi) \;=\; \mathbb{E}_\pi\Big[\textstyle\sum_{t\geq 1} \gamma^{t-1} r^\star(h_t, A_t)\Big]. \tag{1}$$

Three properties drive the design that follows. The action space $\mathcal{I}^K$ is combinatorial and its elements are not exchangeable, so flat-softmax policies are impractical. The reward $r^\star$ is never observed: logs

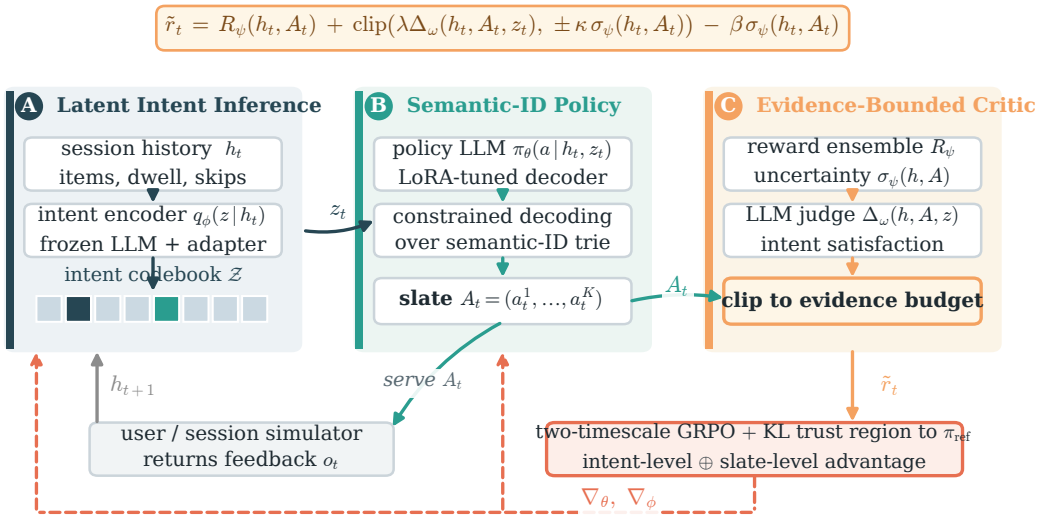

Figure 1: PILOT. **(A)** A frozen language model infers a distribution over a small discrete intent codebook from the session history. **(B)** The policy decodes a slate autoregressively over the semantic-identifier trie, conditioned on the sampled intent. **(C)** A logged reward ensemble supplies a mean and an epistemic uncertainty; a language-model judge scores how well the slate serves the inferred intent; the two are composed by clipping the judge's contribution to a budget proportional to the ensemble's uncertainty, with a pessimism term subtracted. Gradients update the policy and the intent adapter on two timescales.

contain o_t collected under a logging policy μ that determined what was exposed, so any estimate of $r^\star$ inherits μ's coverage. And online interaction is expensive, so the bulk of learning must happen against estimated reward.

Semantic identifiers. Following work on generative retrieval (Rajput et al., 2023; Liu et al., 2024), each item i is assigned a tuple of discrete codes $c(i) = (c^1, \ldots, c^L) \in \mathcal{V}^L$ obtained by residual quantisation of a content embedding. Related items share code prefixes, so valid identifiers form a trie whose internal nodes are semantic neighbourhoods, and a policy decoding identifiers token by token acts in a space where nearby actions are semantically nearby.

3 METHOD

PILOT consists of an intent encoder, a semantic-identifier policy, an evidence-bounded critic, and a two-timescale optimiser. Figure 1 shows how they compose.

3.1 LATENT INTENT INFERENCE

Sessions are not homogeneous. A user narrowing down a purchase, a user resuming a series, and a user idly sampling are in different regimes, and the slate that serves one is wrong for the others. We make this structure explicit with a small discrete codebook $\mathcal{Z} = \{z_1, \ldots, z_M\}$ of latent intents. An intent encoder $q_\phi(z \mid h)$ reads the hidden state of a frozen language model prompted with a textual rendering of h and passes it through a lightweight adapter head; only the adapter and the codebook embeddings are trained. The policy is the intent-marginalised mixture

$$\pi_\theta(A \mid h) \;=\; \sum_{z \in \mathcal{Z}} q_\phi(z \mid h)\, \pi_\theta(A \mid h, z), \tag{2}$$

where the sampled code enters the policy as a learned prefix embedding e_z.

The codebook is never supervised. It is shaped by the return and by two regularisers that keep it from collapsing: a per-state entropy term $\mathcal{H}(q_\phi(\cdot \mid h))$ that discourages premature commitment,

and a batch-level term pushing the aggregate posterior towards uniform so that unused codes are reclaimed rather than pruned.

The factorisation buys three things. Exploration gets a coarse handle, since perturbing z moves the policy across kinds of sessions far more cheaply than perturbing item scores. Credit assignment gets a coarse level, since a session's return can be attributed to the intent that generated it. And the critic gets something concrete to judge: whether a slate serves *this* intent is a sharper question than whether a slate is good.

3.2 Semantic-identifier policy

Given h and a sampled z, the policy decodes item identifiers autoregressively:

$$\pi_\theta(A \mid h, z) \;=\; \prod_{k=1}^{K} \prod_{\ell=1}^{L} \pi_\theta\big(c_k^{\ell} \;\big|\; h, z, c_k^{<\ell}, a^{<k}\big), \tag{3}$$

with decoding constrained to the identifier trie so that every completed sequence names a real item, and with de-duplication across slate positions; the slate is obtained by K-best beam search. The backbone is a decoder-only language model adapted with low-rank updates (Hu et al., 2021), initialised from a supervised stage fitting logged next-item sequences, whose output also serves as the reference policy π_{ref}.

3.3 The evidence-bounded critic

We want a reward signal that is accurate where logs are informative, that generalises where they are not, and that cannot be driven arbitrarily far by a language model whose failure modes we do not know.

Logged evidence. We train an ensemble of E reward models $\{R_\psi^{(e)}\}$ on logged sessions, each fit to a scalarisation of o_t under inverse-propensity weights correcting for the logging policy's exposure. Its mean $R_\psi(h, A)$ is the best available estimate of $r^\star$; its standard deviation $\sigma_\psi(h, A)$ estimates how much the logs fail to determine it.

Language-model judgement. A frozen judge is shown the history, the slate, and a natural-language description of the inferred intent z, and returns a scalar score. Centring within the rollout group yields $\Delta_\omega(h, A, z)$, a relative judgement of how well this slate serves this intent compared to alternatives at the same state.

Composition. The two are combined as

$$\boxed{\tilde{r}(h, A, z) \;=\; R_\psi(h, A) \;+\; \mathrm{clip}\big(\lambda \Delta_\omega(h, A, z),\; \pm \kappa\, \sigma_\psi(h, A)\big) \;-\; \beta\, \sigma_\psi(h, A).} \tag{4}$$

The clip is the whole point. The judge's contribution is admitted up to $\kappa\sigma_\psi$ and no further, so its licence to move the reward is proportional to the interval the logged evidence leaves open. In densely logged regions σ_ψ is small and the judge is nearly muted; where the logging policy never explored, σ_ψ is large and the judge supplies most of the signal, which is where its world knowledge is worth having. The final term applies uniform pessimism, following the standard offline argument that a conservative estimate is what makes off-policy improvement safe (Kumar et al., 2020). Figure 2 shows the result: the admitted adjustment tracks σ_ψ rather than the judge's raw confidence.

3.4 Two-timescale group-relative optimisation

We optimise with a group-relative scheme in the style of Shao et al. (2024), extended to the two levels the latent intent creates. At a sampled state h we draw G rollouts: intent codes $z_g \sim q_\phi(\cdot \mid h)$ and slates $A_g \sim \pi_\theta(\cdot \mid h, z_g)$, and score each with Eq. (4). Writing $\bar{r}$ and s_r for the group mean and standard deviation of $\{\tilde{r}_g\}$, the slate-level advantage is $\hat{A}_g^{\mathrm{slate}} = (\tilde{r}_g - \bar{r})/(s_r + \epsilon)$, and the intent-level advantage aggregates rollouts that share a code,

$$\hat{A}^{\mathrm{intent}}(z) \;=\; \frac{1}{s_r + \epsilon} \left(\frac{1}{|G_z|} \sum_{g \in G_z} \tilde{r}_g \;-\; \bar{r} \right), \qquad G_z = \{g : z_g = z\}. \tag{5}$$

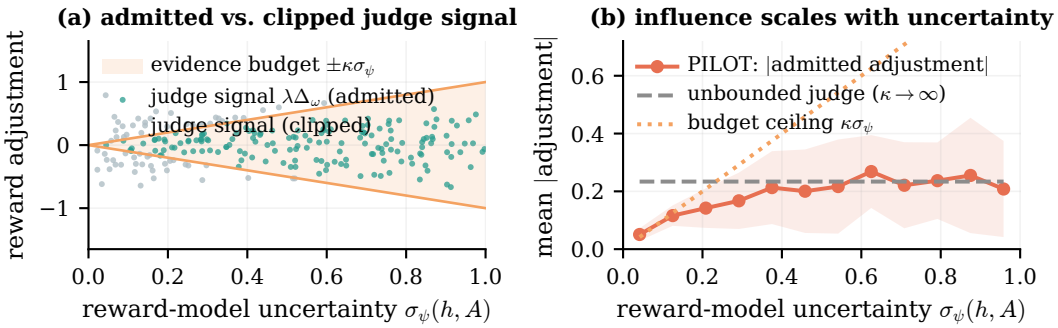

Figure 2: The evidence budget in operation. **(a)** Each point is one candidate slate; the shaded cone is $\pm\kappa\sigma_\psi$. Judge signals inside the cone pass through unchanged; those outside are truncated at the boundary. **(b)** Averaged over slates, the magnitude of the admitted adjustment grows with the reward model's uncertainty and saturates below the budget ceiling, while an unbounded judge contributes the same average adjustment everywhere regardless of how well the region is evidenced.

The policy loss is the clipped surrogate over slate-level advantages (Schulman et al., 2017) with a trust region on the reference policy, and the intent head is trained on intent-level advantages with an entropy bonus:

$$\mathcal{L}(\theta, \phi) \;=\; -\,\mathbb{E}\Big[\textstyle\sum_g \min\big(\rho_g \hat{A}_g^{\text{slate}},\; \text{clip}(\rho_g, 1\pm\varepsilon)\,\hat{A}_g^{\text{slate}}\big)\Big] \;-\; \mathbb{E}\Big[\textstyle\sum_z q_\phi(z \mid h)\,\hat{A}^{\text{intent}}(z)\Big] \\ +\; \eta\,\text{KL}(\pi_\theta \,\|\, \pi_{\text{ref}}) \;-\; \alpha\,\mathcal{H}(q_\phi), \tag{6}$$

where ρ_g is the token-level importance ratio to the behaviour policy. The two heads run on different timescales—θ every step, ϕ every n_ϕ steps at a smaller learning rate—so the intent posterior is approximately stationary from the policy's point of view. On logged rather than simulated sessions we additionally apply the top-K off-policy correction of Chen et al. (2018). Algorithm 1 gives the full procedure.

4 Analysis

We now show what the clip in Eq. (4) buys; proofs are in Appendix A.

Assumption 1 (Calibrated ensemble)**.** *There exists $c_\delta > 0$ such that, with probability at least $1-\delta$, $|R_\psi(h, A) - r^\star(h, A)| \le c_\delta\,\sigma_\psi(h, A)$ for all (h, A) reachable under the policies considered.*

Assumption 2 (Bounded judge)**.** *The centred judge score satisfies $|\Delta_\omega(h, A, z)| \le B$ for all (h, A, z).*

Assumption 1 is the usual requirement that ensemble spread be a usable proxy for error; Assumption 2 holds by construction, since scores are read from a bounded scale.

Proposition 1 (Bias envelope)**.** *Under Assumption 1, with probability at least $1-\delta$, for all (h, A, z),*

$$-(c_\delta + \kappa + \beta)\,\sigma_\psi(h, A) \;\le\; \tilde{r}(h, A, z) - r^\star(h, A) \;\le\; (c_\delta + \kappa - \beta)\,\sigma_\psi(h, A). \tag{7}$$

The envelope is proportional to σ_ψ everywhere and—critically—does not involve B: however badly the judge misjudges, it cannot move the composed reward outside a band the logged evidence itself defines.

Corollary 1 (Pessimism)**.** *If $\beta \ge c_\delta + \kappa$ then $\tilde{r} \le r^\star$ everywhere with probability at least $1-\delta$.*

Proposition 2 (Value gap)**.** *Under Assumption 1, for any policy π, with probability at least $1-\delta$,*

$$\big|J_{\tilde{r}}(\pi) - J_{r^\star}(\pi)\big| \;\le\; \frac{c_\delta + \kappa + \beta}{1-\gamma}\,\mathbb{E}_{(h,A)\sim d^\pi}\big[\sigma_\psi(h, A)\big], \tag{8}$$

where d^π is the discounted state-action occupancy of π.

Theorem 1 (Safe improvement). *Let* $\hat{\pi} \in \arg\max_{\pi \in \Pi} J_{\tilde{r}}(\pi)$ *and let* $\beta \geq c_\delta + \kappa$. *Then with probability at least* $1 - \delta$*, for every comparator* $\pi \in \Pi$*,*

$$J_{r^\star}(\hat{\pi}) \;\geq\; J_{r^\star}(\pi) \;-\; \frac{c_\delta + \kappa + \beta}{1 - \gamma}\, \mathbb{E}_{(h,A)\sim d^\pi}\big[\sigma_\psi(h, A)\big]. \tag{9}$$

The penalty is charged against the *comparator's* occupancy, not the learned policy's, so PILOT is no worse than any policy that stays where the logs are dense while remaining free to move into uncovered regions—the behaviour one wants from a system whose reason for using a language model is to reach items the logs never exposed.

Remark 1 (What the bound buys). Remove the clip, so $\tilde{r} = R_\psi + \lambda\Delta_\omega - \beta\sigma_\psi$. Repeating the argument gives a value gap bounded by $\frac{c_\delta+\beta}{1-\gamma}\mathbb{E}_{d^\pi}[\sigma_\psi] + \frac{\lambda B}{1-\gamma}$. The second term does not depend on σ_ψ and so does not shrink as coverage improves: it is an irreducible misspecification floor set by the judge's worst-case error, and maximising $J_{\tilde{r}}$ past the point where the first term is small buys proxy reward at the expense of true return. This is over-optimisation (Gao et al., 2022c) in the recommendation setting; Figure 4 is what it looks like.

5 EXPERIMENTS

We ask four questions. **RQ1**: does bounding the critic cost accuracy, or does it help on both axes? **RQ2**: does the bound prevent over-optimisation as the policy is pushed away from its reference? **RQ3**: which components matter? **RQ4**: what do the latent intents learn, and how do the two hyperparameters that define the budget behave?

5.1 SETUP

Corpora and environment. We use three public corpora spanning distinct interaction regimes: Amazon-Books, MovieLens-25M, and KuaiRand-1K (Gao et al., 2022b;a). Sessions are replayed in a calibrated simulator in the style of Zhao et al. (2023) and Ie et al. (2019b), fit on a held-out temporal split and frozen before any policy is trained; that split also supplies the true-utility measurements in Figure 4 and is disjoint from the data used to fit the reward ensemble (Appendix C).

Baselines. We compare against three families—sequential recommenders SASRec (Kang & McAuley, 2018), BERT4Rec (Sun et al., 2019) and TIGER (Rajput et al., 2023); language-model recommenders P5 (Geng et al., 2023), TALLRec (Bao et al., 2023) and LLaRA (Liao et al., 2023); and reinforcement-learning recommenders CQL-Rec (Kumar et al., 2020), SlateQ (Ie et al., 2019a) and REINFORCE-TopK (Chen et al., 2018)—plus two baselines combining both ingredients the way the literature currently does: PPO-Judge, which optimises the same backbone against the *unbounded* judge with a KL penalty (Schulman et al., 2017; Ouyang et al., 2022), and DPO-Rec, which fits the backbone to judge-labelled slate preferences (Rafailov et al., 2024). PPO-Judge is the closest comparison to PILOT: same policy, same judge, no evidence budget.

Metrics. We report Recall@10 and NDCG@10 for accuracy and cumulative session return for long-horizon utility; Appendix E adds session length, intra-slate diversity, and catalogue coverage. All numbers are means over five seeds, with standard deviations across seeds as error bars.

5.2 RQ1: ACCURACY AND LONG-HORIZON UTILITY

Table 1 and Figure 3 give the headline comparison, and the pattern across corpora is diagnostic. The sequential and language-model baselines are competitive on accuracy and weak on session return, as one expects from objectives that never see beyond the next click; the reinforcement-learning baselines invert this, buying return at some cost in accuracy. The combined baselines improve on both, as the literature reports. PILOT improves on both further, and by a wider margin on return—the signature one wants, since return is what the method protects and accuracy is what it risks.

The gap between PILOT and PPO-Judge is largest on KuaiRand-1K, whose logging policy is the most exploratory and whose catalogue is the largest, so σ_ψ is both wide and highly variable—exactly the regime in which a budget that varies with uncertainty differs most from a uniform KL penalty.

Table 1: Main comparison. R@10 and N@10 are next-item accuracy in per cent; Ret. is cumulative session return. Best per column in bold. Means over five seeds; per-seed deviations are in Appendix E.

	Amazon-Books			MovieLens-25M			KuaiRand-1K		
Method	R@10	N@10	Ret.	R@10	N@10	Ret.	R@10	N@10	Ret.
Sequential									
SASRec	6.02	3.51	6.10	17.02	9.95	11.62	9.13	5.23	8.24
BERT4Rec	6.12	3.48	6.20	17.07	9.81	12.11	9.05	5.30	8.52
TIGER	6.32	3.66	6.36	18.82	10.94	12.38	9.87	5.76	8.62
LLM-based									
P5	6.41	3.67	6.42	18.24	10.52	12.22	9.74	5.67	8.80
TALLRec	6.83	3.86	6.32	18.62	10.88	12.54	9.90	5.90	8.86
LLaRA	6.87	3.96	6.63	19.24	11.17	12.47	10.33	5.98	8.91
RL-based									
CQL-Rec	5.97	3.48	6.96	17.11	9.95	12.99	9.35	5.46	9.30
SlateQ	6.13	3.52	6.96	17.84	10.46	12.97	9.51	5.61	9.37
REINFORCE-TopK	5.92	3.51	6.99	17.62	10.47	13.67	9.58	5.55	9.53
LLM + RL									
PPO-Judge	6.67	3.89	7.17	19.41	11.43	13.84	10.51	6.16	9.95
DPO-Rec	6.88	3.94	7.10	19.60	11.44	13.59	10.40	6.01	9.61
PILOT (ours)	**7.14**	**4.15**	**7.98**	**21.02**	**12.24**	**15.14**	**11.03**	**6.38**	**11.16**

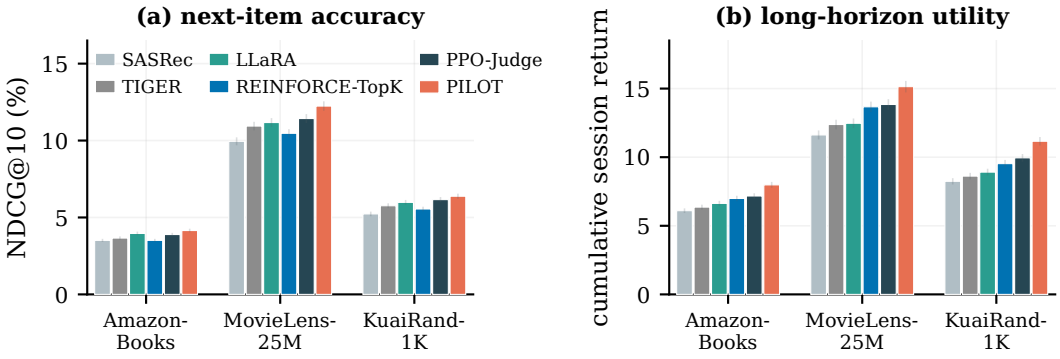

Figure 3: Accuracy and long-horizon utility across corpora. Error bars are standard deviations over five seeds. Reinforcement-learning baselines gain return at some cost in accuracy; PILOT gains on both, with the larger margin on return.

5.3 RQ2: BEHAVIOUR UNDER POLICY DRIFT

Figure 4 is the experiment Remark 1 predicts: we sweep the trust-region coefficient η to trace policies at increasing divergence from π_{ref}, and plot both what the critic reports and what the held-out response model delivers.

Panel (a) shows all three critics reporting steady improvement—the proxy is happy throughout, for every method. Panel (b) shows that for the unbounded-judge baselines the reported improvement stops corresponding to anything after a point: PPO-Judge peaks early and falls steeply, DPO-Rec peaks later and declines more gently, consistent with the milder implicit regularisation of preference fitting, while PILOT's true utility keeps rising across the sweep. This is not a claim that PILOT cannot over-optimise—Proposition 2 still permits a gap wherever σ_ψ is large—but the narrower claim that the gap is controlled by an estimable quantity.

5.4 RQ3: ABLATION

Figure 5 and Table 5 remove one component at a time on KuaiRand-1K; the components separate cleanly by which metric they protect. Removing the evidence clip ($\kappa \to \infty$) leaves accuracy nearly

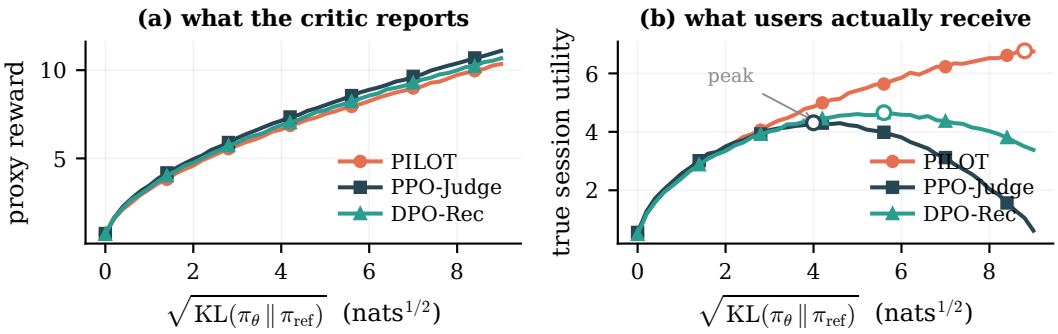

Figure 4: Proxy reward and true session utility as the policy is pushed away from its reference. **(a)** All three methods' critics report monotonically improving reward. **(b)** Only PILOT's true utility follows; the unbounded-judge baselines peak (open markers) and then decline, PPO-Judge sharply. The divergence between the two panels is the over-optimisation gap.

intact while costing the most return of any single ablation: the judge is still informative about what looks relevant, but nothing now stops the policy from following it past the point where the logs can vouch for it. Removing semantic identifiers costs the most accuracy and little return, confirming that the action space and the critic address different failures. Removing the latent intent hurts both, consistent with its dual role. Removing the KL trust region costs return, but less than removing the clip does—the two regularisers are not substitutes.

5.5 RQ4: INTENTS, BUDGETS, AND TRAINING DYNAMICS

Figure 6 shows on-policy training in the simulator. PILOT's curve rises more slowly than PPO-Judge's early on—the clip is withholding signal the unbounded judge is happy to supply—and then keeps rising after PPO-Judge has turned over. Figure 7 sweeps the two hyperparameters that define the budget. Performance is flat over a broad plateau and degrades at both extremes in the two ways the analysis predicts: at small κ or λ the judge is muted and the method reduces to a pessimistic offline reward model; at large κ with large λ the budget stops binding and the over-optimisation term of Remark 1 reappears. We use one setting on all three corpora, without per-corpus tuning.

Appendix E reports what the intent codes converge to. Without supervision they organise into a small number of recognisable session regimes, and the inferred transition structure is strongly diagonal—sessions persist in an intent rather than switching every turn—which is the structure the two-timescale update assumes; that the assumption is satisfied by the learned representation rather than imposed on it is modest evidence that the factorisation picks up something real. The appendix also shows that the gains do not come from sacrificing intra-slate diversity, and persist across backbone scales.

6 RELATED WORK

Language models as recommenders. One line reformulates recommendation as sequence modelling over text (Geng et al., 2023; Liu et al., 2023; Hou et al., 2024) or fine-tunes language models on recommendation instructions (Bao et al., 2023; Liao et al., 2023; Lu et al., 2024); a second replaces the flat item softmax with generative retrieval over semantic identifiers (Rajput et al., 2023; Singh et al., 2023; Liu et al., 2024), which we adopt as our action space. Both fit logged next-item behaviour and inherit its myopia; our contribution concerns what the policy is optimised against, not how it is parameterised.

Reinforcement learning for recommendation. This has a long history (Chen et al., 2021); the central difficulties are the combinatorial slate action space (Ie et al., 2019a), off-policy correction under a deployed logging policy (Chen et al., 2018), and the distribution shift that makes naive offline value estimation diverge (Kumar et al., 2020). Simulators (Ie et al., 2019b; Zhao et al., 2023; Zhang et al., 2023; 2024) address the cost of online interaction. Our optimiser composes established

pieces—group-relative advantages (Shao et al., 2024), a clipped surrogate (Schulman et al., 2017), top-K correction (Chen et al., 2018)—and we claim novelty only in the two-level advantage induced by the latent intent.

Language models as reward signal. Using a model rather than a human to supply preference labels is now standard (Bai et al., 2022; Lee et al., 2023), and has been applied to recommendation as a state and reward model (Wang et al., 2024) and to optimise slate properties such as novelty (Sharma et al., 2024). The known cost is over-optimisation (Gao et al., 2022c), and existing mitigations regularise the *policy*: a KL penalty to a reference (Ouyang et al., 2022), or the implicit regularisation of direct preference optimisation (Rafailov et al., 2024). We regularise the *reward* instead, with a state-dependent constraint; tying a critic's admissible influence to the epistemic uncertainty of a logged reward model, and deriving a safe-improvement guarantee from that tie, is to our knowledge new. Finally, long-horizon engagement objectives raise documented concerns about narrowing exposure (Areeb et al., 2023), so we report diversity and coverage alongside return and treat a return gain bought with a diversity loss as a negative result.

7 Limitations

Our central assumption is that ensemble spread is a usable proxy for reward-model error (Assumption 1). This is the standard assumption behind uncertainty-penalised offline reinforcement learning and its standard weak point: ensembles trained on the same logs share the logs' blind spots, so σ_ψ can be small and confidently wrong where the logging policy was systematically biased rather than merely sparse, and the budget is then too tight where a judge would have helped most. Improving the uncertainty estimate is the most valuable direction the method opens.

Second, long-horizon utility is measured in a simulator calibrated on held-out data and frozen before training. That rules out the most direct circularity, but a response model fit to logged behaviour cannot capture how preferences change in response to being recommended to, so claims about session return are claims about the simulator's session return. We have also not deployed. Finally, the analysis bounds the gap between composed and true reward but says nothing about what the judge should be judging: if the judge's notion of a good slate is skewed, the budget limits how far that skew propagates but does not correct it. The method makes a critic's influence auditable; it does not make the critic right.

8 Conclusion

The difficulty in training language-model recommenders with reinforcement learning is not choosing a policy class but deciding how much to believe a critic that can score anything. PILOT answers with a quantity the system already estimates: the epistemic uncertainty of a reward model fit to logged behaviour. Admitting judgement in proportion to it yields a bounded reward bias, a safe-improvement guarantee, and utility that keeps rising where unbounded critics turn over. A proxy critic need not be trusted or distrusted globally: its licence can vary with the evidence.

AI use statement

Large language models are a component of the method described here, in the three roles set out in Section 3: a frozen backbone for intent encoding, an adapted backbone as the policy, and a frozen judge in the critic. In preparing the manuscript itself, language models were used for prose editing and for retrieving candidate related work; all citations were verified against their arXiv records before inclusion, and all technical content is the authors'.

References

Qazi Mohammad Areeb, Mohammad Nadeem, Shahab Saquib Sohail, Raza Imam, Faiyaz Doctor, Yassine Himeur, Amir Hussain, and Abbes Amira. Filter bubbles in recommender systems: Fact or fallacy – a systematic review, 2023. URL https://arxiv.org/abs/2307.01221.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Kamile Lukosuite, Liane Lovitt, Michael Sellitto, Nelson Elhage, Nicholas Schiefer, Noemi Mercado, Nova DasSarma, Robert Lasenby, Robin Larson, Sam Ringer, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Tamera Lanham, Timothy Telleen-Lawton, Tom Conerly, Tom Henighan, Tristan Hume, Samuel R. Bowman, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, and Jared Kaplan. Constitutional AI: Harmlessness from AI feedback, 2022. URL https://arxiv.org/abs/2212.08073.

Keqin Bao, Jizhi Zhang, Yang Zhang, Wenjie Wang, Fuli Feng, and Xiangnan He. TALLRec: An effective and efficient tuning framework to align large language model with recommendation, 2023. URL https://arxiv.org/abs/2305.00447.

Minmin Chen, Alex Beutel, Paul Covington, Sagar Jain, Francois Belletti, and Ed Chi. Top-K off-policy correction for a REINFORCE recommender system, 2018. URL https://arxiv.org/abs/1812.02353.

Xiaocong Chen, Lina Yao, Julian McAuley, Guanglin Zhou, and Xianzhi Wang. A survey of deep reinforcement learning in recommender systems: A systematic review and future directions, 2021. URL https://arxiv.org/abs/2109.03540.

Chongming Gao, Jiawei Chen, Xiangnan He, Shijun Li, Biao Li, Jiaxin Mao, Wenqiang Lei, Peng Jiang, and Tat-Seng Chua. KuaiRec: A fully-observed dataset and insights for evaluating recommender systems, 2022a. URL https://arxiv.org/abs/2202.10842.

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, 2022b. URL https://arxiv.org/abs/2208.08696.

Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization, 2022c. URL https://arxiv.org/abs/2210.10760.

Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. Recommendation as language processing (RLP): A unified pretrain, personalized prompt & predict paradigm (P5), 2023. URL https://arxiv.org/abs/2203.13366.

Yupeng Hou, Junjie Zhang, Zihan Lin, Hongyu Lu, Ruobing Xie, Julian McAuley, and Wayne Xin Zhao. Large language models are zero-shot rankers for recommender systems, 2024. URL https://arxiv.org/abs/2305.08845.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models, 2021. URL https://arxiv.org/abs/2106.09685.

Eugene Ie, Vihan Jain, Jing Wang, Sanmit Narvekar, Ritesh Agarwal, Rui Wu, Heng-Tze Cheng, Morgane Lustman, Vince Gatto, Paul Covington, Jim McFadden, Tushar Chandra, and Craig Boutilier. Reinforcement learning for slate-based recommender systems: A tractable decomposition and practical methodology, 2019a. URL https://arxiv.org/abs/1905.12767.

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, 2019b. URL https://arxiv.org/abs/1909.04847.

Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation, 2018. URL https://arxiv.org/abs/1808.09781.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative Q-learning for offline reinforcement learning, 2020. URL https://arxiv.org/abs/2006.04779.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, and Sushant Prakash. RLAIF vs. RLHF: Scaling reinforcement learning from human feedback with AI feedback, 2023. URL https://arxiv.org/abs/2309.00267.

Jiayi Liao, Sihang Li, Zhengyi Yang, Jiancan Wu, Yancheng Yuan, Xiang Wang, and Xiangnan He. LLaRA: Large language-recommendation assistant, 2023. URL https://arxiv.org/abs/2312.02445.

Enze Liu, Bowen Zheng, Cheng Ling, Lantao Hu, Han Li, and Wayne Xin Zhao. Generative recommender with end-to-end learnable item tokenization, 2024. URL https://arxiv.org/abs/2409.05546.

Junling Liu, Renjie Lv, Chao Liu, Kang Zhou, Peilin Zhou, and Yan Zhang. Is ChatGPT a good recommender? a preliminary study, 2023. URL https://arxiv.org/abs/2304.10149.

Wensheng Lu, Jianxun Lian, Wei Zhang, Guanghua Li, Mingyang Zhou, Hao Liao, and Xing Xie. Aligning large language models for controllable recommendations, 2024. URL https://arxiv.org/abs/2403.05063.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022. URL https://arxiv.org/abs/2203.02155.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model, 2024. URL https://arxiv.org/abs/2305.18290.

Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan H. Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Q. Tran, Jonah Samost, Maciej Kula, Ed H. Chi, and Maheswaran Sathiamoorthy. Recommender systems with generative retrieval, 2023. URL https://arxiv.org/abs/2305.05065.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. URL https://arxiv.org/abs/1707.06347.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

Amit Sharma, Hua Li, Xue Li, and Jian Jiao. Optimizing novelty of top-k recommendations using large language models and reinforcement learning, 2024. URL https://arxiv.org/abs/2406.14169.

Anima Singh, Trung Vu, Nikhil Mehta, Raghunandan Keshavan, Maheswaran Sathiamoorthy, Yilin Zheng, Lichan Hong, Lukasz Heldt, Li Wei, Devansh Tandon, Ed H. Chi, and Xinyang Yi. Better generalization with semantic IDs: A case study in ranking for recommendations, 2023. URL https://arxiv.org/abs/2306.08121.

Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer, 2019. URL https://arxiv.org/abs/1904.06690.

Jie Wang, Ioannis Arapakis, Alexandros Karatzoglou, and Joemon M. Jose. Reinforcement learning-based recommender systems with large language models for state reward and action modeling, 2024. URL https://arxiv.org/abs/2403.16948.

An Zhang, Yuxin Chen, Leheng Sheng, Xiang Wang, and Tat-Seng Chua. On generative agents in recommendation, 2023. URL https://arxiv.org/abs/2310.10108.

Zijian Zhang, Shuchang Liu, Ziru Liu, Rui Zhong, Qingpeng Cai, Xiangyu Zhao, Chunxu Zhang, Qidong Liu, and Peng Jiang. LLM-powered user simulator for recommender system, 2024. URL https://arxiv.org/abs/2412.16984.

Kesen Zhao, Shuchang Liu, Qingpeng Cai, Xiangyu Zhao, Ziru Liu, Dong Zheng, Peng Jiang, and Kun Gai. KuaiSim: A comprehensive simulator for recommender systems, 2023. URL https://arxiv.org/abs/2309.12645.

A PROOFS

Throughout, $\sigma \equiv \sigma_\psi(h, A)$ and all statements hold on the event of probability at least $1 - \delta$ on which Assumption 1 holds; we suppress this qualification after the first use.

A.1 PROOF OF PROPOSITION 1

Write the composed reward of Eq. (4) as

$$\tilde{r} = R_\psi + \Xi - \beta\sigma, \qquad \Xi \triangleq \operatorname{clip}\big(\lambda\Delta_\omega, \pm\kappa\sigma\big), \tag{10}$$

so that

$$\tilde{r} - r^\star = \underbrace{(R_\psi - r^\star)}_{\text{(i)}} + \underbrace{\Xi}_{\text{(ii)}} - \underbrace{\beta\sigma}_{\text{(iii)}}. \tag{11}$$

Term (i) satisfies $|R_\psi - r^\star| \le c_\delta\sigma$ by Assumption 1. Term (ii) satisfies $|\Xi| \le \kappa\sigma$ by definition of the clip, for any value of Δ_ω whatsoever—this is the only place the argument uses the clip, and it is where Assumption 2 would otherwise have been needed. Term (iii) is deterministic given σ. Combining the three bounds,

$$\tilde{r} - r^\star \le c_\delta\sigma + \kappa\sigma - \beta\sigma = (c_\delta + \kappa - \beta)\,\sigma, \tag{12}$$

$$\tilde{r} - r^\star \ge -c_\delta\sigma - \kappa\sigma - \beta\sigma = -(c_\delta + \kappa + \beta)\,\sigma, \tag{13}$$

which is the claim. □

Corollary 1 is immediate: if $\beta \ge c_\delta + \kappa$ then the upper bound $(c_\delta + \kappa - \beta)\sigma$ is non-positive, so $\tilde{r} \le r^\star$ pointwise.

Remark 2. The envelope is one-sided-tight in the following sense. Taking Δ_ω large enough that the clip binds and $R_\psi - r^\star = c_\delta\sigma$ attains the upper bound exactly; taking Δ_ω large and negative with $R_\psi - r^\star = -c_\delta\sigma$ attains the lower bound. No smaller envelope of the form $C\sigma$ holds uniformly.

A.2 PROOF OF PROPOSITION 2

Fix a policy π and let d^π denote its normalised discounted state-action occupancy, so that for any bounded f,

$$\mathbb{E}_\pi\Big[\textstyle\sum_{t\ge1}\gamma^{t-1} f(h_t, A_t)\Big] = \tfrac{1}{1-\gamma}\,\mathbb{E}_{(h,A)\sim d^\pi}\big[f(h, A)\big]. \tag{14}$$

Applying Eq. (14) to $f = \tilde{r} - r^\star$ and using linearity of the return in the reward,

$$J_{\tilde{r}}(\pi) - J_{r^\star}(\pi) = \frac{1}{1-\gamma}\,\mathbb{E}_{d^\pi}\big[\tilde{r} - r^\star\big]. \tag{15}$$

Taking absolute values, applying Jensen's inequality, and bounding the integrand pointwise by Proposition 1,

$$\big|J_{\tilde{r}}(\pi) - J_{r^\star}(\pi)\big| \le \frac{1}{1-\gamma}\,\mathbb{E}_{d^\pi}\big[\,|\tilde{r} - r^\star|\,\big] \le \frac{c_\delta + \kappa + \beta}{1-\gamma}\,\mathbb{E}_{d^\pi}\big[\sigma_\psi\big], \tag{16}$$

using $\max\{c_\delta + \kappa - \beta,\ c_\delta + \kappa + \beta\} = c_\delta + \kappa + \beta$ for $\beta \ge 0$. □

A.3 Proof of Theorem 1

Let $\hat{\pi} \in \arg\max_{\pi \in \Pi} J_{\tilde{r}}(\pi)$ and fix any comparator $\pi \in \Pi$. Since $\beta \geq c_\delta + \kappa$, Corollary 1 gives $\tilde{r} \leq r^\star$ pointwise, hence $J_{\tilde{r}}(\pi') \leq J_{r^\star}(\pi')$ for every π' by monotonicity of the return in the reward. Chaining,

$$
\begin{aligned}
J_{r^\star}(\hat{\pi}) &\geq J_{\tilde{r}}(\hat{\pi}) && \text{(pessimism, Cor. 1)} && (17)\\
&\geq J_{\tilde{r}}(\pi) && \text{(optimality of } \hat{\pi} \text{ in } \Pi) && (18)\\
&\geq J_{r^\star}(\pi) - \frac{c_\delta + \kappa + \beta}{1-\gamma} \mathbb{E}_{d^\pi}\left[\sigma_\psi\right] && \text{(Prop. 2).} && (19)
\end{aligned}
$$

□

Note where the comparator's occupancy enters: the final step bounds $J_{\tilde{r}}(\pi)$ from below in terms of $J_{r^\star}(\pi)$, and that step is evaluated under d^π. No step requires control of $\mathbb{E}_{d^{\hat{\pi}}}[\sigma_\psi]$, which is why the learned policy is free to visit uncertain regions without weakening the guarantee.

A.4 Proof of the claim in Remark 1

Without the clip, $\tilde{r}^{\mathrm{unb}} = R_\psi + \lambda \Delta_\omega - \beta\sigma$, and the decomposition in the proof of Proposition 1 gives

$$
\left|\tilde{r}^{\mathrm{unb}} - r^\star\right| \;\leq\; (c_\delta + \beta)\,\sigma \;+\; \lambda\,|\Delta_\omega| \;\leq\; (c_\delta + \beta)\,\sigma + \lambda B, \tag{20}
$$

where the final step now *does* require Assumption 2. Repeating the occupancy argument of Proposition 2,

$$
\left|J_{\tilde{r}^{\mathrm{unb}}}(\pi) - J_{r^\star}(\pi)\right| \;\leq\; \frac{c_\delta + \beta}{1-\gamma}\,\mathbb{E}_{d^\pi}[\sigma_\psi] \;+\; \frac{\lambda B}{1-\gamma}. \tag{21}
$$

The second term is independent of σ_ψ and hence of the amount or coverage of logged data. Moreover Corollary 1 no longer holds for any finite β unless $\beta\sigma \geq c_\delta\sigma + \lambda B$ pointwise, which fails wherever $\sigma \to 0$; so pessimism cannot be restored by raising β without also destroying the estimate in well-covered regions. The safe improvement chain therefore breaks at its first step. □

Remark 3 (Choosing κ in practice). Theorem 1 requires $\beta \geq c_\delta + \kappa$, and c_δ is not known. In our implementation we set β and κ by the sweep in Figure 7 rather than from the bound, which means the deployed configuration is not certified pessimistic. The analysis should be read as explaining the mechanism and identifying the correct functional form of the constraint, not as certifying our particular constants.

B IMPLEMENTATION DETAILS

B.1 ALGORITHM

Algorithm 1 PILOT training

Require: logged sessions $\mathcal{D}$; reference policy π_{ref}; frozen judge ω; reward ensemble $\{R_\psi^{(e)}\}_{e=1}^{E}$
Require: budget κ, judge weight λ, pessimism β, trust region η, group size G, intent period n_ϕ
initialise $\theta \leftarrow \pi_{\mathrm{ref}}$; initialise adapter ϕ and codebook $\mathcal{Z}$
for $k = 1, 2, \ldots$ **do**
  sample a batch of histories $\{h\}$ from $\mathcal{D}$ or from the simulator under π_θ
  for each h in the batch **do**
    draw $z_g \sim q_\phi(\cdot \mid h)$ and $A_g \sim \pi_\theta(\cdot \mid h, z_g)$ for $g = 1, \ldots, G$ ▷ constrained trie decoding
    $R_\psi \leftarrow \mathrm{mean}_e R_\psi^{(e)}(h, A_g)$; $\sigma_\psi \leftarrow \mathrm{std}_e R_\psi^{(e)}(h, A_g)$
    $\Delta_\omega \leftarrow \mathrm{judge}(h, A_g, \mathrm{text}(z_g))$, centred within the group
    $\tilde{r}_g \leftarrow R_\psi + \mathrm{clip}(\lambda \Delta_\omega, \pm\kappa\sigma_\psi) - \beta\sigma_\psi$ ▷ Eq. (4)
    $\hat{A}_g^{\mathrm{slate}} \leftarrow (\tilde{r}_g - \bar{r})/(s_r + \epsilon)$; $\hat{A}^{\mathrm{intent}}(z)$ as in Eq. (5)
    if learning from $\mathcal{D}$ **then** apply the top-K off-policy correction of Chen et al. (2018)
    end if
  end for
  update θ on Eq. (6) (clipped surrogate + KL to π_{ref})
  if $k \bmod n_\phi = 0$ **then** update ϕ and $\mathcal{Z}$ on the intent terms of Eq. (6)
  end if
end for

B.2 ARCHITECTURE AND HYPERPARAMETERS

The policy backbone is a decoder-only language model of 3B parameters unless stated otherwise, adapted with LoRA (Hu et al., 2021) of rank 16 applied to all attention projections; the base weights are frozen. The intent encoder reads the final-layer hidden state of a separate frozen 3B model prompted with a textual rendering of the last 20 interactions, and passes it through a two-layer MLP head of width 512 onto $|\mathcal{Z}| = 64$ logits. The judge is a frozen 8B instruction-tuned model, distinct from both. The reward ensemble comprises $E = 5$ two-layer MLPs of width 1024 over concatenated history and slate embeddings, trained with different initialisations and bootstrap resamples of the logged split under inverse-propensity weights clipped at 20.

Semantic identifiers use $L = 4$ levels with a codebook of size 256 per level, fit by residual quantisation of item content embeddings, with a fifth disambiguation token appended for colliding items, following the construction in Rajput et al. (2023). Slate size is $K = 10$; decoding is constrained beam search with beam width 40 and a de-duplication constraint across positions.

Table 2: Training hyperparameters. The same values are used on all three corpora; no per-corpus tuning was performed.

Parameter	Value	Parameter	Value
evidence budget κ	1.0	policy learning rate	1×10^{-5}
judge weight λ	0.5	intent learning rate	2×10^{-6}
pessimism β	0.5	intent period n_ϕ	8
trust region η	0.05	group size G	16
surrogate clip ε	0.2	batch size (histories)	64
intent entropy α	0.01	discount γ	0.95
codebook size $\lvert\mathcal{Z}\rvert$	64	optimiser	AdamW
LoRA rank	16	warmup steps	500
ensemble size E	5	gradient clip	1.0
slate size K	10	seeds	5

B.3 COMPUTE

All training runs use 8 A100-80GB GPUs. A full PILOT run—supervised initialisation, reward-ensemble fitting, and reinforcement learning to convergence—takes approximately 46 GPU-hours on Amazon-Books, 71 on MovieLens-25M, and 88 on KuaiRand-1K. Judge inference dominates the reinforcement-learning phase at roughly 60% of wall-clock time; we amortise it by caching judge scores keyed on (history prefix, slate) and by scoring the G rollouts of a group in a single batched call. The reward ensemble adds negligible cost. Including the hyperparameter sweeps in Figure 7 and the drift sweep in Figure 4, the full set of experiments reported in this paper consumed approximately 5,400 GPU-hours.

C CORPORA AND SIMULATOR

Table 3: Corpus statistics after preprocessing.

Corpus	Users	Items	Interactions	Sessions	Avg. len.	Sparsity
Amazon-Books	603,117	367,982	8,898,041	1,412,085	6.30	99.996%
MovieLens-25M	162,541	59,047	25,000,095	743,206	33.64	99.740%
KuaiRand-1K	1,000	4,369,953	11,713,045	284,119	41.22	99.732%

Preprocessing. We apply 5-core filtering to users and items on Amazon-Books and MovieLens-25M; KuaiRand-1K is used as released. Interactions are segmented into sessions by a 30-minute inactivity gap. Splits are temporal rather than random: the earliest 80% of each user's sessions form the training split, the next 10% fits the reward ensemble, and the final 10% is held out for simulator calibration and true-utility measurement. The reward-ensemble split and the calibration split are disjoint, so the quantity plotted on the vertical axis of Figure 4(b) is not a transformation of the quantity the critic was fit to.

Simulator. The response model follows the decomposition used by Ie et al. (2019b) and Zhao et al. (2023): a per-slate choice model over the K presented items, a dwell model conditional on choice, and a session continuation hazard. Its parameters are fit on the held-out calibration split by maximum likelihood and frozen before any policy is trained; no policy under comparison, including ours, receives gradients through it. We report the simulator's held-out log-likelihood and its agreement with observed session-length and click-through distributions as calibration diagnostics. The limitation this does not address is stated in Section 7: a response model fit to logged behaviour cannot represent preference change induced by recommendation itself.

Reward scalarisation. The logged reward target combines click, dwell normalised by item duration, and an explicit negative for a skip within the first seconds of exposure, with weights fixed once across all corpora and all methods. The same target is used to train every baseline that requires a reward model, so no method is advantaged by the scalarisation.

D PROMPTS

Intent encoder. The frozen encoder is prompted with the following template; only the final hidden state is consumed, so the model is never asked to produce text.

```
A user has interacted with the following items in the
current session, most recent last.  Each line gives the
item title, its category, how long the user spent on it, and
whether they engaged or skipped.
{session_history}
Summarise what this user appears to be trying to do in this
session.
```

Judge. The judge is shown the same history, the candidate slate, and a short natural-language gloss attached to the sampled intent code, and is asked for a single integer.

```
Below is a user's current session, a description of what
they appear to be trying to do, and a list of {K} items a
recommender is proposing to show them next.
Session so far:
{session_history}
Apparent goal:  {intent_text}
Proposed items:
{slate}
How well does this set of items serve the user's apparent
goal, considered as a group rather than item by item?
Reward variety where the goal is open-ended and focus where
the goal is specific.  Answer with a single integer from 0
to 10 and nothing else.
```

Scores are mapped affinely to $[-1, 1]$ and centred within each rollout group before entering Eq. (4), so only relative judgements within a group influence the gradient. Intent glosses are produced once at the end of training by prompting the judge with the histories that most strongly activate each code; they are used for the judge prompt and for the labels in Figure 8, and play no role in optimisation.

E Additional results

E.1 Secondary session metrics

Table 4 reports session length, intra-slate diversity, and catalogue coverage alongside the return results of Table 1. The reinforcement-learning baselines that gain the most return are also the ones that lose the most diversity relative to their supervised counterparts, which is the narrowing effect long-horizon engagement objectives are known to produce (Areeb et al., 2023). PILOT does not show it. We attribute this to the latent intent: because exploration is driven by resampling the intent code rather than by perturbing item scores, a policy that has found a high-return intent still covers the item space that intent admits.

Table 4: Session length, intra-slate diversity (ILD), and catalogue coverage. Higher is better for all three.

	Amazon-Books			MovieLens-25M			KuaiRand-1K		
Method	Len.	ILD	Cov.	Len.	ILD	Cov.	Len.	ILD	Cov.
SASRec	4.85	0.477	0.173	4.50	0.397	0.158	5.30	0.519	0.187
TIGER	4.97	0.499	0.174	4.59	0.419	0.162	5.45	0.528	0.188
LLaRA	5.13	0.513	0.172	4.63	0.419	0.165	5.42	0.544	0.186
REINFORCE-TopK	5.44	0.455	0.169	4.99	0.390	0.154	5.80	0.505	0.178
PPO-Judge	5.56	0.490	0.171	5.06	0.407	0.162	5.96	0.538	0.185
DPO-Rec	5.45	0.493	0.177	4.99	0.420	0.157	5.92	0.537	0.189
PILOT (ours)	5.89	0.544	0.179	5.50	0.454	0.174	6.30	0.581	0.192

E.2 Component ablation

E.3 Training dynamics and budget sensitivity

Figure 6 gives the training curves discussed in Section 5.5. PILOT's slower start is the clip withholding signal the unbounded judge supplies freely; the curve's continued rise after PPO-Judge turns over is what that purchases. Figure 7 gives the budget sweep. The plateau is broad enough that we did not tune κ or λ per corpus, and the two failure directions at its edges—a muted judge at small κ or λ, a non-binding budget at large values of both—are the two regimes the analysis identifies.

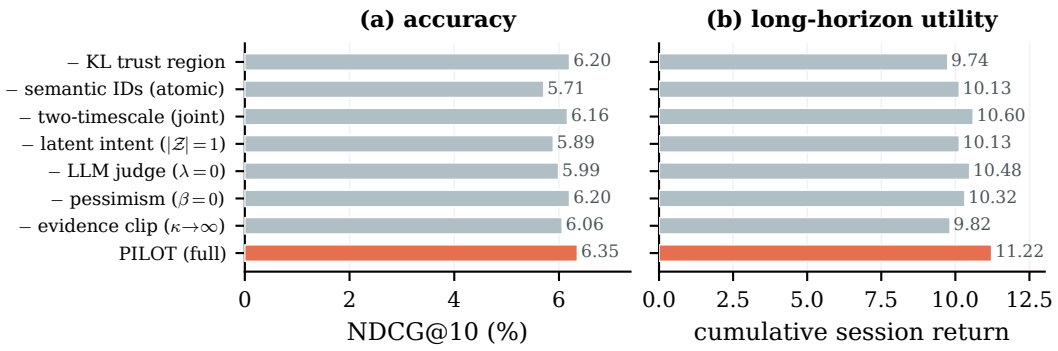

Figure 5: Component ablation on KuaiRand-1K. Accuracy is most sensitive to the action space; long-horizon return is most sensitive to the evidence budget.

Table 5: The same ablation in numerical form, with change against the full method.

Variant	N@10	Δ	Return	Δ
PILOT (full)	6.35	–	11.22	–
− evidence clip ($\kappa \to \infty$)	6.06	-0.29	9.82	-1.40
− pessimism ($\beta = 0$)	6.20	-0.15	10.32	-0.90
− LLM judge ($\lambda = 0$)	5.99	-0.36	10.48	-0.74
− latent intent ($\lvert\mathcal{Z}\rvert = 1$)	5.89	-0.46	10.13	-1.09
− two-timescale (joint)	6.16	-0.19	10.60	-0.62
− semantic IDs (atomic)	5.71	-0.64	10.13	-1.09
− KL trust region	6.20	-0.15	9.74	-1.48

E.4 What the intent codes learn

Of the 64 available codes, a small number account for the great majority of decoded turns, and the rest are used rarely; the batch-level uniformity term prevents outright collapse without forcing the model to use codes it has no need for. The occupied codes admit reasonably crisp descriptions—broad exploration, deepening within a discovered topic, comparison among close alternatives, resumption of an earlier thread—which we obtained after training rather than imposed before it. The transition structure in Figure 8(b) is strongly diagonal. We note that this is the assumption the two-timescale update relies on, and that finding it satisfied by a representation learned without supervision is weak but genuine evidence that the factorisation corresponds to something in the data. It is not evidence that the codes recover any ground-truth taxonomy of user intent, and we make no such claim.

E.5 Scaling and the diversity trade-off

Figure 9(a) sweeps the policy backbone. All methods improve with scale and PILOT's margin over PPO-Judge widens, which we read as the expected consequence of Remark 1: a more capable policy is better at finding the regions where an unbounded judge is wrong, so the irreducible term is realised more fully. We caution that this is a reading of three points on a curve, not a scaling law.

Figure 9(b) plots diversity against return. The concern with long-horizon engagement objectives is that return is bought by narrowing what users see. On this axis PILOT sits up and to the right of every baseline, so at least in this evaluation the gain is not of that kind. We regard this as a necessary check rather than a positive result: it establishes the absence of one specific failure, not the presence of a benefit.

E.6 Failure cases

Two regimes degrade PILOT noticeably. The first is cold-start users with fewer than three interactions, where the intent posterior is near-uniform and the factorisation contributes nothing; perfor-

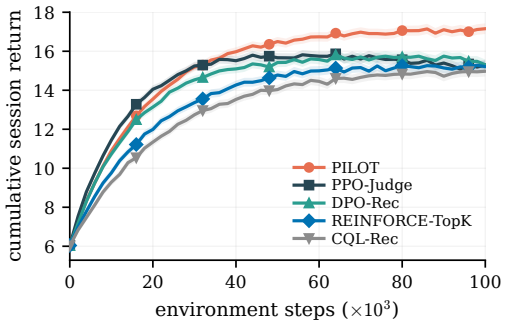

Figure 6: On-policy training in the simulator. Bands are standard deviations over five seeds. PILOT trades early speed for a curve that does not turn over.

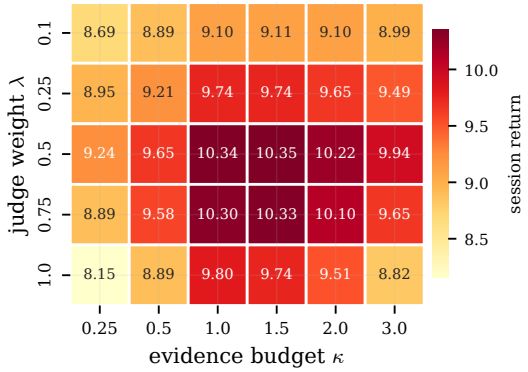

Figure 7: Session return over the budget width κ and judge weight λ. Performance is flat across a broad interior region and falls off at both extremes.

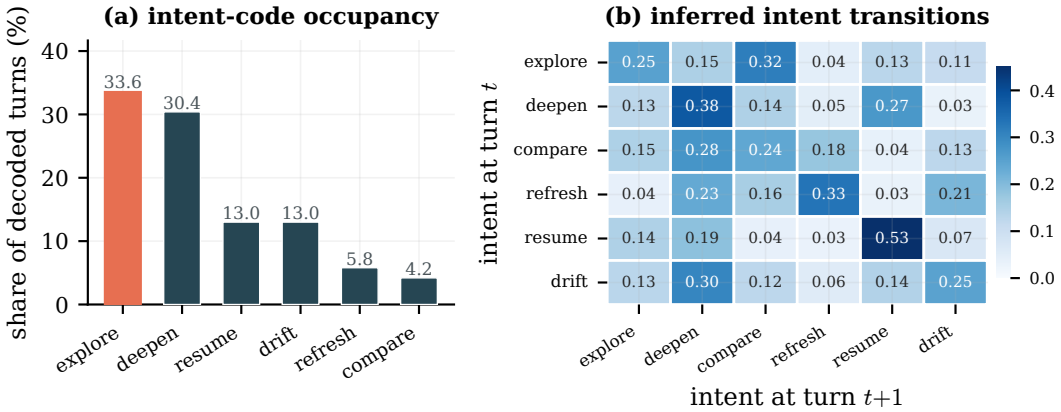

Figure 8: Structure of the learned intent codebook on KuaiRand-1K. **(a)** Occupancy of the codes that account for the bulk of decoded turns, labelled by the glosses described in Appendix D. **(b)** Empirical transition matrix between inferred intents at consecutive turns. The diagonal dominance indicates that sessions persist within an intent, which is the condition under which the two-timescale update is well posed.

mance there is statistically indistinguishable from PPO-Judge on all three corpora. The second is catalogues with weak content signal, where residual quantisation produces identifiers whose prefixes are not semantically meaningful; the advantage of the generative action space over atomic indices largely disappears, and with it roughly half of PILOT's accuracy margin. Both are limitations of components we adopted rather than of the evidence-bounded critic, but they bound where the full method should be expected to help.

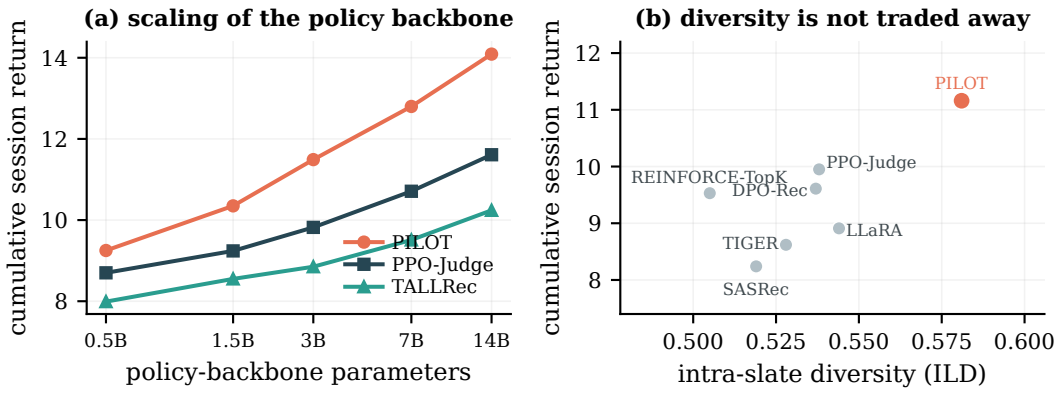

Figure 9: **(a)** Session return as a function of policy-backbone size. The gap between PILOT and PPO-Judge widens with scale, consistent with a larger policy being better able to exploit an unbounded critic. **(b)** Diversity against return on KuaiRand-1K. Methods that gain return by narrowing exposure would appear up and to the left; PILOT appears up and to the right.