GROUND THE PAST TO PREDICT THE FUTURE: OPTIMIZING NATURAL-LANGUAGE USER REPRESENTATIONS WITHOUT DOWNSTREAM SUPERVISION

Anonymous authors
Paper under double-blind review

ABSTRACT

In search and recommendation systems, user behavioral logs such as search queries and item clicks have long served as the substrate for user modeling. Typically, these interaction histories are distilled into dense representations, such as user embeddings trained end-to-end for the downstream task they serve. More recently, advances in large language models (LLMs) have opened a second axis for user representation in the form of natural-language user profiles, textual summaries of a user's behaviors. However, when checked against the input behaviors they are meant to summarize, LLM-generated profiles often misrepresent the user, overstating or over-inferring beyond what the evidence shows. This same interpretability means a profile's quality can be measured by whether it faithfully abstracts past behavior, a training signal that requires no downstream task at all. It is also a natural hypothesis that such faithfulness to the past would help predict the future. Yet existing methods still optimize profiles using downstream supervision from future interactions, leaving this hypothesis untested. We study this question by operationalizing groundedness as a reward along three axes (cohesion, alignment, and truthfulness) and training a profile generator against it with group-wise Direct Preference Optimization (DPO), without any downstream supervision. Across three behavioral log datasets, including a proprietary industry dataset, and multiple personalized search and recommendation tasks, groundedness-optimized profiles consistently improve over the base model and perform comparably to those trained with downstream supervision, ranking best or second-best in 13 of 16 settings. Groundedness further complements downstream supervision when the two rewards are combined. These results suggest that faithfulness to past behavior can serve as an effective training signal for user representations.

1 INTRODUCTION

User behavioral logs (e.g., queries, clicks, dwell time, and anything else users leave behind) from search and recommendation systems are one of the most abundant sources of signal for understanding users, and thus offer a valuable substrate for user modeling (Purificato et al., 2024; Li & Zhao, 2021). Traditionally, this substrate has been distilled into dense vector representations, fixed-size embeddings that encode a user's interests and are optimized end-to-end for whatever downstream task they serve (Kang & McAuley, 2018; Sun et al., 2019). Increasingly, however, with advances in large language models (LLMs), this signal is instead distilled into natural-language user representations, interpretable summaries that describe a user in text (Ren et al., 2024; Zhang, 2024). This shift opens up a new way of optimizing and evaluating a user representation. A dense vector is uninterpretable by construction, so downstream usefulness has naturally been the single criterion by which its quality is judged. However, a natural-language user representation admits a further criterion. Since it can be read and checked against the input behavior sequence that produced it, we can directly assess whether it faithfully reflects the past user behavior it is meant to summarize, or instead overstates, hallucinates, or omits parts of it, even before it is put to any downstream use.

However, most work on learning natural-language user representations does not exploit this new optimization axis. Instead, it continues to treat the representation as a means to an end, optimizing it directly against a downstream objective, such as item rating or engagement prediction (Chen

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.

et al., 2025b; Shi et al., 2025b; Wang et al., 2025b; Gao et al., 2025b). Yet past groundedness and future predictability are intuitively connected. A representation that faithfully captures what a user is like should be useful for anticipating what the user will want next. Conversely, an overstated or hallucinated profile misrepresents the user (see Appendix A.1 for examples), and a downstream model then treats these distortions as genuine signal. To our knowledge, this connection remains largely untested. In this paper, we examine whether user representations optimized to be grounded in past behavior transfer to downstream personalization on future interactions. We study this question in implicit feedback settings such as clicks, which are far more abundant in real-world systems, whereas prior profile-learning methods have been evaluated largely on explicit feedback such as item ratings or reviews (Hou et al., 2026; Harper & Konstan, 2015).

We propose optimizing natural-language user representations for groundedness, the degree to which a representation is a cohesive, well-aligned, and truthful abstraction of a user's observed behaviors. Concretely, we represent a user as a set of profiles, each induced by grouping related past behaviors together and abstracting them (Section 2). We turn groundedness into a training reward along three dimensions that follow from the structure of a profile, an evidence cluster paired with a profile text describing it: (i) how cohesively a profile's supporting behaviors group together, (ii) how well its profile text aligns with that behavior group, and (iii) how truthfully the profile text represents the group without overstating or hallucinating beyond it. We train a generative language model (LLM) against this reward via a groupwise Direct Preference Optimization (DPO) framework (Rafailov et al., 2023; Liu et al., 2025; Leng et al., 2026), with no downstream objective in the loop (Section 3). We experiment on three user behavioral log datasets from search and recommendation that differ in domain: a proprietary multi-domain service log (Srv.), a shopping log (MerRec) (Li et al., 2025), and a web search log (AOL4PS) (Guo et al., 2021) (Section 4.1). For each dataset, we evaluate the groundedness-optimized profiles on its supported downstream tasks (Section 4.4), and compare them against profiles optimized directly for user engagement prediction, following the paradigm of prior work (Section 4.3). We find that groundedness-optimized profiles match profiles trained directly against a downstream objective over future interactions, across multiple dataset–task evaluations (Section 5). The contributions of this work are summarized as follows:

- We propose groundedness, faithfulness to observed behavior, as an optimization axis for natural-language user representations, in contrast to the prevailing paradigm of downstream optimization.
- We turn groundedness into a reward signal along three dimensions, (i) cohesion, (ii) alignment, and (iii) truthfulness, and train a user profile generation model against it via groupwise Direct Preference Optimization (DPO), analyzing its effect on the resulting profiles.
- Across three behavioral log datasets and multiple personalized search and recommendation tasks, groundedness-optimized profiles perform comparably to those trained directly against a downstream objective, consistently improving over the base model.

2 Task Formulation

We consider the problem of learning natural-language user representations from search and recommendation behavioral logs that are useful as user features for downstream personalization tasks. For a user u, let $L_u^{\text{past}} = (\ell_1, \ldots, \ell_n)$ denote a past behavioral log, a sequence of raw interaction events (e.g., search, click) within a fixed time window, from which we induce a user representation. We hold out a future window $L_u^{\text{future}} = (\ell_{n+1}, \ldots, \ell_{n+k})$ to evaluate how well this representation helps downstream models anticipate the user's subsequent interactions.

User Behavioral Units. Unlike explicit user feedback data such as item ratings or reviews (Hou et al., 2026; Harper & Konstan, 2015), where each unit is a single, deliberate statement of user preference or interest, behavioral logs are inherently implicit, noisy, and fragmented. We therefore first consolidate L_u^{past} into a set of meaningful behavioral units

$$B_u^{\text{past}} = f(L_u^{\text{past}}) = (b_1, \ldots, b_{n'}), \quad n' \leq n, \tag{1}$$

using an off-the-shelf aggregation procedure f, which is not part of our optimization. Each unit b_i aggregates one or more temporally or topically related raw interaction events in L_u^{past} into a single coherent behavioral units (e.g., viewing several similar items multiple times is consolidated into a single unit b_i that captures this sequence as reflecting a single underlying interest). Further details and examples of raw-interaction-to-behavioral-unit consolidation are provided in Appendix A.3.

Natural-Language User Profile Generation. A generative language model π takes B_u^{past}, the set of behavioral units from the input window, as input. A prompt function $\Phi(\cdot)$ formats B_u^{past} into an instruction, and π produces a set of user profiles in a single pass

$$P_u = \pi\big(\Phi(B_u^{\text{past}})\big) = \{p_1, \ldots, p_k\}, \tag{2}$$

Each profile $p_i = (E_i, y_i)$ consists of an evidence cluster $E_i \subseteq B_u^{\text{past}}$, a subset of behavioral units grouped together as related, and a profile text y_i inducing what E_i reveals about the user. It is y_i itself that constitutes the natural-language representation of that aspect of the user. The resulting user profile is the collection of these texts, $Y_u = \{y_1, \ldots, y_k\}$.

Downstream Evaluation. The quality of a generated user profile Y_u is ultimately judged by how useful it is when consumed by downstream models for personalization tasks. Given a candidate pool C_u, which contains the items in L_u^{future} as ground-truth targets alongside distractor items, and a ranking function g, we rank C_u using Y_u as the user representation,

$$\hat{C}_u = g(Y_u, C_u), \tag{3}$$

and score the resulting ranking $\hat{C}_u$ against these ground-truth targets using ranking metrics. Since g is a design choice, we compare multiple instantiations to assess whether the usefulness of Y_u generalizes across them. We apply this formulation to the downstream tasks supported by each dataset, each testing whether Y_u promotes the items in L_u^{future} to a higher rank within a task-specific C_u. Full definitions of C_u, g, evaluation metrics, and each task for every dataset are in Section 4.4.

3 METHOD

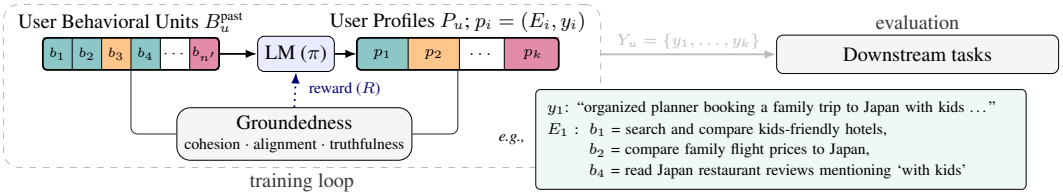

Figure 1: Overview of our framework. Behavioral units $B_u^{\text{past}} = \{b_1, \ldots, b_{n'}\}$ observed within a fixed time window arrive as json text, and the model π emits json text parsed into profiles $P_u = \{p_1, \ldots, p_k\}$, where each $p_i = (E_i, y_i)$ pairs an evidence cluster $E_i \subseteq B_u^{\text{past}}$ with a profile text y_i. Matching colors indicate which units support which profile, and the box on the right gives an example. Groundedness scores P_u against B_u^{past} along cohesion, alignment, and truthfulness, and serves as the training reward. Downstream tasks lie outside it, consuming only the profile texts $Y_u = \{y_1, \ldots, y_k\}$ as the user representation to rank a candidate pool against the items in L_u^{future}.

Overview. We propose a groundedness reward-based training framework that optimizes a model π to induce evidence-grounded profiles from a user's behavioral units $B_u^{\text{past}} = \{b_1, \ldots, b_{n'}\}$ (Figure 1). Given B_u^{past}, the model outputs a set of profiles $P_u = \{p_1, \ldots, p_k\}$, where each profile $p_i = (E_i, y_i)$ pairs an evidence cluster $E_i \subseteq B_u^{\text{past}}$ with a text representation y_i. Internally, y_i consists of a title and D supporting descriptions,

$$y_i = \big(\text{title}_i, \{\text{desc}_{i,j}\}_{j=1}^{D}\big), \tag{4}$$

where the title names the profile and each description provides supporting detail.

Starting from a model's initial policy π_0, we train π_θ so that generated profiles exhibit higher groundedness with respect to their supporting evidence. Specifically, we encourage (i) cohesive evidence cluster E_i, where grouped behavioral units form semantically consistent groups, (ii) strong profile–evidence alignment, where y_i abstracts the shared trait of the assigned E_i, and (iii) truthfulness, where y_i does not hallucinate beyond the observed evidence in E_i. In addition, we introduce two regularizers to prevent excessive omission of B_u^{past} and to discourage overly small or large cluster E_i. We discuss the design rationale behind this decomposition in Appendix A.1.

Profiles Sampling. For a given input B_u^{past}, we generate a group of training rollouts by sampling n candidate profile sets $\hat{P}_u^{(1)}, \ldots, \hat{P}_u^{(n)}$ from an initial policy π_0:

$$\hat{P}_u^{(1)}, \ldots, \hat{P}_u^{(n)} \sim \pi(\cdot \mid B_u^{\text{past}}). \tag{5}$$

Groundedness Reward. Given a rollout P_u[1], we score each profile $p_i \in P_u$ along three groundedness components: cohesion, alignment, and truthfulness, each capturing a different aspect of how faithfully a profile reflects its evidence. First, cohesion score $s_{\text{coh}}(p_i)$ combines (i) intra-evidence cohesion between behavioral units within the same cluster (μ_{in}), along with the variance of that similarity as a penalty for inconsistency (σ_{in}), and (ii) separation from outer behavioral units (μ_{cross}). Here, C_{ab} denotes the semantic similarity between behavioral unit a and b[2], and $\bar{E}_i$ denotes the units in B_u^{past} not assigned to profile p_i:

$$\mu_{\text{in}} = \mathbb{E}_{a \neq b,\, a,b \in E_i}[C_{ab}]; \quad \sigma_{\text{in}} = \text{Std}_{a \neq b,\, a,b \in E_i}[C_{ab}]; \quad \mu_{\text{cross}} = \mathbb{E}_{a \in E_i,\, b \in \bar{E}_i}[C_{ab}]$$

$$s_{\text{coh}}(p_i) = (\mu_{\text{in}} - \lambda \sigma_{\text{in}}) + (\mu_{\text{in}} - \mu_{\text{cross}})$$

Second, alignment score $s_{\text{align}}(p_i)$ measures how well y_i aligns with the content of E_i. Third, truthfulness score $s_{\text{truth}}(p_i)$ evaluates whether y_i is strictly grounded in E_i without overstatement. We compute the alignment and truthfulness scores using LLM-based judges[3], J_{align} and J_{truth}, respectively, which take (y_i, E_i) as input and output calibrated scores in $[0, 1]$:

$$s_{\text{align}}(p_i) = J_{\text{align}}(y_i, E_i); \qquad s_{\text{truth}}(p_i) = J_{\text{truth}}(y_i, E_i) \tag{6}$$

We combine these three components into a single per-profile groundedness score:

$$s_{\text{ground}}(p_i) = \text{mean}\big(s_{\text{align}}(p_i),\, s_{\text{truth}}(p_i),\, s_{\text{coh}}(p_i)\big). \tag{7}$$

Taken together, $s_{\text{ground}}(p_i)$ reflects whether the profile text y_i faithfully captures the user's underlying behaviors, without overstating or hallucinating beyond what was actually observed.

Reward Aggregation and Structural Regularizer. In addition to the groundedness reward above, we introduce two regularizers on the size of E_i and coverage of P_u: a cluster size regularizer $s_{\text{size}}(E_i)$ and a behavior coverage regularizer $s_{\text{cov}}(P_u; B_u^{\text{past}})$. First, the size score $s_{\text{size}}(E_i)$ softly bounds evidence-cluster sizes to $[\tau_{\text{size}}^{\min}, \tau_{\text{size}}^{\max}]$, discouraging overly fine- or coarse-grained clusters:

$$s_{\text{size}}(E_i) = \begin{cases} 1 - \left(\frac{\tau_{\text{size}}^{\min} - |E_i|}{\tau_{\text{size}}^{\min} - 1}\right)^2 & \text{if } |E_i| < \tau_{\text{size}}^{\min} \\ 1 & \text{if } \tau_{\text{size}}^{\min} \leq |E_i| \leq \tau_{\text{size}}^{\max} \\ 1 - \left(\frac{|E_i| - \tau_{\text{size}}^{\max}}{\tau_{\text{size}}^{\max}}\right)^2 & \text{if } |E_i| > \tau_{\text{size}}^{\max} \end{cases}$$

Second, to encourage the profiles to collectively form a comprehensive representation of B_u^{past}, we define a coverage ratio over the full output P_u,

$$s_{\text{cov}}(P_u; B_u^{\text{past}}) = \frac{\left|\bigcup_{p_i \in P_u} E_i\right|}{\left|B_u^{\text{past}}\right|},$$

and apply a soft threshold, capping $\tilde{s}_{\text{cov}}$ at 1 once $s_{\text{cov}} \geq \tau_{\text{cov}}$ and leaving it unchanged otherwise. Unlike $s_{\text{size}}(E_i)$, which is defined per evidence cluster, $s_{\text{cov}}(P_u; B_u^{\text{past}})$ is defined at the output level, over the entire set P_u.

We combine $s_{\text{ground}}(p_i)$ and $s_{\text{size}}(E_i)$ into a profile-level reward,

$$r(p_i) = \alpha_1 \cdot s_{\text{ground}}(p_i) + \alpha_2 \cdot s_{\text{size}}(E_i), \tag{8}$$

and aggregate profile-level rewards together with the output-level coverage regularizer into the final groundedness reward:

$$R(P_u) = \alpha_3 \cdot \frac{1}{|P_u|} \sum_{p_i \in P_u} r(p_i) + \alpha_4 \cdot \tilde{s}_{\text{cov}}, \tag{9}$$

where $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ are scalar weights controlling the contribution of each component. Note that R is the reward that optimizes for groundedness; it is a function of P_u and B_u^{past} alone, and requires no access to L_u^{future} during training. The values we use for all hyperparameters introduced above (λ, $\tau_{\text{size}}^{\min}$, $\tau_{\text{size}}^{\max}$, τ_{cov}, α_1–α_4) are provided in Appendix A.6.

[1]For brevity, we drop the candidate index (g) and write $P_u = \hat{P}_u^{(g)}$ for an arbitrary candidate.

[2]The computation details of unit similarity are provided in Section 4.2.

[3]Full judge prompts are in Appendix A.8.2; further details, including judge reliability, are in Appendix A.4.

Optimization. We optimize π against the groundedness reward R via groupwise DPO (Liu et al., 2025; Leng et al., 2026), an extension of Direct Preference Optimization (DPO) (Rafailov et al., 2023) that converts the rewards of all n rollouts $\{\hat{P}_u^{(1)}, \ldots, \hat{P}_u^{(n)}\}$ into a soft preference distribution and optimizes π against the whole group at once, learning to favor the traits of higher-reward profiles. Full objectives and further training details are provided in Appendix A.6.

4 EXPERIMENT

4.1 DATASETS

We experiment on three user behavioral log datasets that differ in domain and in what is logged: (i) a proprietary industry dataset from a nationwide online platform (`Srv.`) and two public datasets, (ii) `MerRec` (Li et al., 2025) and (iii) `AOL4PS` (Guo et al., 2021). `Srv.` covers two services, search and a recommendation feed, spanning multiple content domains. Its search log has queries, impressions, and clicks, and its feed log has impressions and clicks. `MerRec` is a shopping recommendation log without impressions, in which each click is tagged with its interaction type (e.g., view, like, or buy). `AOL4PS` is a web search log with queries, impressions, and clicks. For each dataset, we sample 10k users and extract an 8-week period of their logs. Table 1 summarizes the datasets. Further details including platform, domain, and item metadata are provided in Appendix A.2.

Table 1: Dataset characteristics and extraction details. INT. TYPES lists the interaction types in each dataset. PERIOD USED denotes the 8-week log period extracted for our experiments, and # USERS, # INT., and # B. UNITS denote the number of sampled users, their total interactions, and the total number of behavioral units within that period, respectively. Behavioral units are derived by processing raw interactions, as defined in Section 2, with construction details provided in Appendix A.3.

Dataset	Data Characteristics			Data Extraction			
	SERVICE	DOMAIN	INT. TYPES	PERIOD USED	# USERS	# INT.	# B. UNITS
`Srv.` [4]	Search & Rec.	Multiple	Search, Click	Jul 13, 2026 – Sep 6, 2026	10,000	28,142,112	6,983,908
`MerRec`	Rec.	Shopping	Click	May 1, 2023 – Jun 25, 2023	10,000	3,673,697	1,151,384
`AOL`	Search	Web	Search, Click	Mar 1, 2006 – Apr 25, 2006	10,000	1,420,662	551,286

We split an 8-week period of logs into training, validation, and test sets using a sliding-window scheme (Figure 2), pairing each two-week input window[5] with the following week as its target window. Downstream-supervised methods use training-split target windows as a reward signal (Section 4.3), whereas ours never accesses them. All methods are evaluated on validation and test target windows. We further partition users into disjoint splits at a 7:1:2 ratio, so that profiles are evaluated on unseen users.

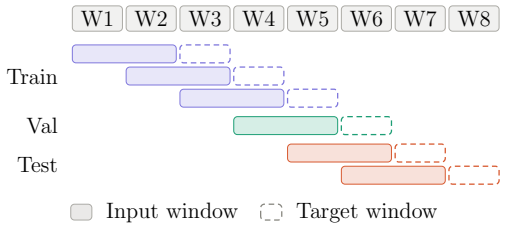

Figure 2: Sliding-window split of the 8-week log period. Note that our method requires no target-window supervision during training.

4.2 MODELS

We use QWEN3.6-35B-A3B (Qwen, 2026) as the backbone for our main experiments, and additionally train GEMMA-4-26B-A4B-IT (Gemma, 2026) to test whether our training method generalize across model families. Behavioral unit similarity C_{ab} (Section 3) is computed as the cosine similarity between BGE-M3 (Chen et al., 2024) embeddings.

[4] The `Srv.` dataset covers only selected services and does not include the platform's complete logs.

[5] The input window length is a design choice that determines the time horizon of the user characteristics the profile represents; we use two weeks in all experiments. The three datasets differ in density, averaging 171.6 behavioral units per input window in `Srv.`, 30.0 in `MerRec`, and 15.5 in `AOL4PS`.

4.3 COMPARED METHODS

We compare against two downstream-supervised methods that reward rollouts using the target window during training (Figure 2), differing in the downstream task used to compute the reward:

- **RLPF** (Wu et al., 2025) optimizes on a next-activity prediction task, training the user profile to be predictive of the user's future activities by prompting an LLM to pick the true next activity from a four-way multiple-choice set (the target activity plus three random negatives).
- **LETTINGO** (Wang et al., 2025b) optimizes on a sentiment classification task, training the user profile to help an LLM correctly classify the user's sentiment toward individual future items.

Their original implementations differ not only in this task definition but also in the backbone model, the LLM used to compute the reward, prompting, and the training algorithm, which confounds direct comparison. To isolate the effect of the reward signal, we standardize the training setup so that all methods, including ours, differ only in the reward function used to score each rollout. Further methodological details and implementation specifics for each method are provided in Appendix A.6.

4.4 EVALUATION

We evaluate resulting user profiles on multiple downstream settings that differ in how the candidate pool C_u and ranking function g are instantiated (Table 2). These settings reflect the two stages common to search and recommendation pipelines (Baeza-Yates et al., 1999; Huang et al., 2025): a broad **retrieval (L1)** setting that ranks a large item pool, and a more fine-grained **ranking (L2)** setting that reorders a small, pre-filtered shortlist. We also evaluate **personalized search (PS)**, which reorders the set of documents impressed for a search query, conditioned on the user's profile.

L1 requires an efficient mechanism to rank a large pool, so g uses bi-encoders to embed profiles and items independently and ranks by embedding similarity. We use MULTILINGUAL-E5-LARGE (Wang et al., 2024) and BGE-M3 (Chen et al., 2024), which can be served as both retrievers and rankers and are therefore applied at every stage. In L2 and PS the pool is already narrowed, making it affordable to score each profile–candidate pair jointly. We therefore additionally evaluate QWEN3-RERANKER-8B (Zhang et al., 2025b), a Qwen3-based pointwise reranker, and LRL (Ma et al., 2023), a prompting-based listwise reranker that generates a reordered list of candidate identifiers in a single pass, which we instantiate with GEMMA-4-31B (Gemma, 2026). Table 3 summarizes the per-dataset evaluation setup.

Table 2: Overview of the three downstream task types and the datasets that support each. C_u denotes the candidate pool and g the ranking function. We instantiate g with MULTILINGUAL-E5-LARGE (E5), BGE-M3 (BGE), QWEN3-RERANKER-8B (Q3-R), and LISTWISE RERANKER WITH LLM (LRL).

Task	C_u	g	**Datasets**
Retrieval (L1)	Full item pool	E5, BGE	`Srv.`, `MerRec`
Ranking (L2)	Item shortlist	E5, Q3-R, LRL	`Srv.`, `MerRec`
Pers. Search (PS)	Impression list	E5, Q3-R, LRL	`Srv.`, `AOL4PS`

L1. Given a user profile, the retrieval task ranks the full items appearing in our dataset (POOL SIZE) and evaluates whether the user's positive item(s) from the target window are successfully retrieved among the pool. We report Recall@K (Schütze et al., 2008).

L2. We construct one ranking slate per user's positive item in the target window. Each positive is paired with 19 hard negatives, forming a fixed ranking candidate set of 20 items[6]. Given the user's profile, the L2 ranking task then predicts a relevance ordering over the slate. We

Table 3: Per-dataset evaluation setup. POOL SIZE: candidate pool size per ranking instance; # POS.: average number of positive items (i.e., items clicked in the target window) per instance.

Dataset	**Task**	POOL SIZE	# POS. (avg.)
`Srv.`	L1	1,131,956	101.9
	L2	20	1
	PS	15.1	1.64
`MerRec`	L1	2,119,722	36.9
	L2	20	1
`AOL4PS`	PS	10	1

[6]Hard negatives should be plausible enough to challenge the ranker rather than trivially distinguishable from the positive: for `Srv.`, drawn from the same user's recommendation impressions; for `MerRec`, which lacks impression logs, constructed via a sampling procedure (Appendix A.2.1).

report NDCG (Järvelin & Kekäläinen, 2002). Since a user may have multiple ranking instances (one per positive item in the target window), we first average scores within each user before averaging across users, so that users with more items do not disproportionately influence the reported metrics.

PS. The candidate set for PS is the search impression list already present in the log, the set of documents that were actually shown to the user for a given search query. Given the user profile, the task reorders the impression list, and we evaluate whether the user clicked document(s) are promoted to a higher rank within it. We report NDCG, as in L2.

5 RESULTS

Table 4 reports downstream evaluation results on the test sets of three datasets. Training for groundedness alone shows a consistent transfer to downstream performance. OURS improves over BASE across datasets and tasks, with statistically significant gains in 15 of 16 settings. Moreover, the resulting profiles are as useful as those directly trained to predict the future interaction, with OURS performing comparably to LETTINGO and RLPF and achieving the best or second-best result in 13 of 16 settings. The gain is most pronounced on L1 retrieval, where OURS improves over BASE by 79%/88% on `MerRec` and 23%/19% on the more challenging `Srv.` (E5/BGE), whose retrieval task involves nearly three times as many positive items per user (101.9 vs. 36.9, Table 3). OURS improves consistently over BASE on L2 ranking and personalized search as well, though, as with LETTINGO and RLPF, the absolute margins are smaller than on L1 retrieval.

Table 4: Test-set performance on the downstream tasks supported by each dataset, comparing user profiles trained with different reward functions under an otherwise identical training pipeline. We report Recall@100 for retrieval (L1) and NDCG for recommendation ranking (L2) and personalized search (PS) (Section 4.4). BASE denotes the backbone model (QWEN3.6-35B-A3B) without any training; + LETTINGO and + RLPF trained with rewards using their respective downstream objectives (Section 4.3), while + OURS trained with groundedness reward alone without any supervision on the target window (Section 3). *, †, and ‡ denote a statistically significant improvement over BASE (paired t-test) at $p < 0.05$, $p < 0.01$, and $p < 0.001$, respectively.

(a) `MerRec` and `AOL4PS`

	MerRec					AOL4PS		
	L1		**L2**			**PS**		
Method	E5	BGE	E5	Q3-R	LRL	E5	Q3-R	LRL
BASE	0.0192	0.0206	0.3699	0.3990	0.4566	0.5297	0.5653	0.6463
+ LETTINGO	0.0253‡	0.0326‡	**0.3736**‡	0.4033‡	**0.4640**‡	0.5419‡	0.5854‡	0.6583‡
+ RLPF	0.0254‡	0.0339‡	0.3731‡	**0.4038**‡	0.4621‡	**0.5458**‡	0.5855‡	**0.6654**‡
+ OURS	**0.0344**‡	**0.0387**‡	0.3725‡	0.4037‡	0.4613‡	0.5413‡	**0.5940**‡	0.6639‡

(b) `Srv.`

	L1		**L2**			**PS**		
Method	E5	BGE	E5	Q3-R	LRL	E5	Q3-R	LRL
BASE	0.0022	0.0036	0.3589	0.3896	0.3949	0.4141	0.4210	0.4897
+ LETTINGO	0.0026†	0.0040	0.3601‡	**0.3918**‡	**0.3958**†	0.4150	0.4220	0.4918†
+ RLPF	0.0022	0.0038	0.3597‡	0.3898	0.3954	0.4152*	0.4208	0.4899
+ OURS	**0.0027**‡	**0.0043**†	**0.3613**‡	0.3910‡	0.3957*	**0.4163**‡	**0.4221**	**0.4997**‡

Validation Tracking During Training To examine this transfer closely, Figure 3 tracks groundedness and downstream task performance on the `MerRec` validation set every 500 training steps. Although training never optimizes for future-facing prediction, downstream performance (Figure 3b–c) broadly rises with groundedness (Figure 3a) across both task types. Among the groundedness components, truthfulness improves the most, followed by cohesion and alignment. L1 retrieval tracks groundedness closely over training, while L2 ranking is noisier but stays above BASE during training across all three rankers, despite never targeting an objective resembling either task.

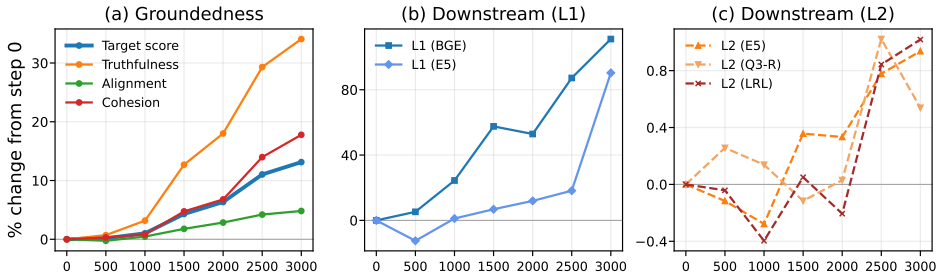

Figure 3: Groundedness and downstream performance on the MerRec validation set across training, reported as % change relative to BASE (the untrained step-0 model) at every 500 steps. (a) Groundedness, shown as its three components, TRUTHFULNESS, ALIGNMENT, and COHESION (Section 3), and their mean (TARGET SCORE). (b) L1 retrieval. (c) L2 ranking. Checkpoints are shown up to step 3,000, which achieves the highest validation groundedness and is used for all test evaluations in Table 4; no downstream signal is used for its selection.

Combining with Downstream Supervision The groundedness reward and the downstream reward target different objectives. Groundedness is retrospective, rewarding a profile for faithfully explaining behavior already observed, while RLPF is prospective, rewarding a profile for predicting behavior. To test whether the two provide complementary signals, we combine them by averaging both scores on the same rollouts (OURS + RLPF) and train on MerRec (Figure 4). The effect splits by task type. On L1 retrieval, groundedness alone already accounts for most of the gain, and adding RLPF brings no further improvement. On L2 ranking, combining the two rewards yields gains beyond either alone across rankers. It suggests that groundedness and downstream reward can serve as complementary training signals.

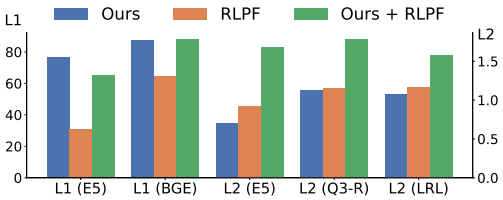

Figure 4: % improvement (L1/L2) over BASE on MerRec for OURS, RLPF, and OURS + RLPF.

Qualitative Analysis Table 5 compares statistical aspects of generated profiles across methods. OURS produces fewer, larger profiles per user than both BASE and RLPF (2.74 vs. 3.93/4.31 profiles; 10.70 vs. 7.28/6.85 units per profile). Although each profile is more elaborate ($|y_i|$ = 58.4 vs. 52.6/50.8 words), the resulting user representation Y_u is shorter because it contains fewer profiles ($|Y_u|$ = 160.2 vs. 206.7/218.7). OURS also rarely induces profiles from a single unit (0.5% vs. 9.5%/10.4%), leaving thin evidence unassigned rather than forcing it into a profile, at a small coverage cost (0.89 vs. 0.93/0.94). This structural behavior is likely driven largely by the cluster-size and coverage regularizers (s_{size}, s_{cov}), since s_{size} directly penalizes clusters below $\tau^{\min}_{\text{size}}$. However, as Section 5 shows, these regularizers alone fail to provide a meaningful training signal. The downstream gains instead come from the groundedness components, which shape how a profile describes the user and make the resulting representation more useful. In comparison, RLPF tends to produce narrower, more fragmented profiles centered on individual interests, consistent with its larger number of profiles and smaller evidence clusters. LETTINGO remains close to BASE on the structural measures, suggesting that its downstream gains come primarily from qualitative changes in profile text rather than structural changes. Combining groundedness with RLPF (O+R) yields profiles structurally between the two, consistent with the complementarity observed in Figure 4. Appendix A.7 provides full case studies (A–E) showing what these numbers look like in practice: OURS declining to build a profile from a single ambiguous unit (A), RLPF splitting a thinly-supported niche interest into its own profile (B), OURS and LETTINGO each describing identical evidence more specifically than BASE (C, D), and OURS+RLPF combining traits of both rewards (E).

Table 5: Statistical comparison of generated profiles over the MerRec test set ($P_u = \{p_1, \ldots, p_k\}$ with $p_i = (E_i, y_i)$, as defined in Section 3). LG denotes LETTINGO; O+R denotes the OURS+RLPF model in Figure 4.

Metric	BASE	RLPF	LG	OURS	O+R
$\lvert P_u\rvert$ (profiles/user)	3.93	4.31	3.96	2.74	3.40
$\lvert E_i\rvert$ (evidence/profile)	7.28	6.85	7.40	10.70	8.51
% ($\lvert E_i\rvert \le 1$)	9.5	10.4	9.1	0.5	3.8
s_{cov} (coverage)	0.93	0.94	0.95	0.89	0.92
$\lvert y_i\rvert$ (words/profile)	52.6	50.8	54.1	58.4	58.2
$\lvert Y_u\rvert$ (total words/user)	206.7	218.7	214.2	160.2	197.7

Reward Ablation To isolate each component's contribution, we retrain OURS on MerRec, removing one groundedness component at a time, and measure the resulting drop from the full reward across the same downstream tasks as in Table 4. Truthfulness is the primary driver on L1 retrieval, erasing 71% (E5) and 85% (BGE) of the gain when removed. Alignment instead drives most of the gain on L2 ranking, erasing 44–49% of the gain on two of three rankers (Q3-R, LRL). Cohesion has a mixed effect throughout, with no consistent pattern across settings. Training with the regularizers alone, without any groundedness component, fails to learn, often performing no better than untrained BASE, indicating that the groundedness axes are responsible for the downstream gains.

Backbone Generalization To examine whether the observed transfer generalizes beyond a single model family, we repeat training with GEMMA-4-26B-A4B-IT on MerRec and AOL4PS. Groundedness training again improves significantly over the untrained model in all settings ($p < 0.001$), indicating that this transfer holds across model families (Appendix A.5).

6 RELATED WORKS

6.1 NATURAL-LANGUAGE USER PROFILES

User representations have typically been dense vectors learned from the items a user interacted with (Covington et al., 2016; Kang & McAuley, 2018), with multi-interest variants representing a user with several vectors, each summarizing a distinct group of behaviors (Li et al., 2019; Cen et al., 2020). Our set of profiles follows the same intuition but expresses each interest as readable text. Natural-language profiles offer two advantages over such embeddings: they are interpretable (Radlinski et al., 2022), and they can be passed directly as context to LLM-based rankers and other LLM-driven components, serving many downstream uses without retraining a task-specific encoder (Ma et al., 2023; Ren et al., 2024). With the advance of LLMs, a line of work generates such user profiles (Xi et al., 2024; Zhang, 2024). What varies most across this work is the signal against which the profile generator is optimized. A common choice is to hold out part of the history and reward a profile by how well a downstream LLM recovers that held-out signal, whether by classifying the user's preference toward an item, predicting the next activity, or ranking the user's true interaction above those of other users (Wang et al., 2025a; Wu et al., 2025; Ju et al., 2026). Other approaches vary the training procedure through iterative refinement, joint training with a downstream ranker, or bandit-based record selection (Chen et al., 2025a; Gao et al., 2025a; Zhang et al., 2025a; Du et al., 2026; Shi et al., 2025a; Sun et al., 2025). In all of these, profiles are judged by downstream utility, and their faithfulness to the behaviors they summarize is left unoptimized.

6.2 FAITHFULNESS IN LANGUAGE GENERATION

Faithfulness to a source is a long-standing concern in text generation, most extensively studied in abstractive summarization, where models often produce content unsupported by the input (Maynez et al., 2020; Ji et al., 2023). Recent work verifies long-form generations claim by claim (Min et al., 2023) or uses LLMs as judges to score outputs at scale (Zheng et al., 2023). Closest to our training setup, Tian et al. (2024) fine-tune with DPO (Rafailov et al., 2023) on preferences built from automatic factuality scores, without human labels. More broadly, reinforcement learning from AI feedback (RLAIF) replaces human labels with LLM judgments, remaining effective even when the judge is the policy model itself (Bai et al., 2022; Lee et al., 2024). Our reward follows this paradigm but applies it to user modeling, where the source to be faithful to is the user's own behavior history.

7 CONCLUSION

In this work, we examined whether grounding natural-language user profiles in past behavior yields representations useful for downstream personalization. We found that profiles trained only for groundedness consistently improved downstream performance over the untrained backbone and performed comparably to methods optimized to predict users' future interactions, across evaluation settings. Combined with downstream supervision, groundedness further benefited fine-grained ranking. These findings suggest that grounding a profile in past behavior is an effective training signal for user representations, motivating future study of how to measure and optimize groundedness.

AI USE STATEMENT

We used generative AI tools (Claude) to assist with language polishing of the manuscript. All AI-assisted text was reviewed and verified by the authors.

REFERENCES

Anthropic. Introducing claude sonnet 5. https://www.anthropic.com/news/claude-sonnet-5, 2026.

Ricardo Baeza-Yates, Berthier Ribeiro-Neto, et al. *Modern information retrieval*, volume 463. ACM press New York, 1999.

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.

Yukuo Cen, Jianwei Zhang, Xu Zou, Chang Zhou, Hongxia Yang, and Jie Tang. Controllable multi-interest framework for recommendation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, pp. 2942–2951, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450379984. doi: 10.1145/3394486.3403344. URL https://doi.org/10.1145/3394486.3403344.

Aili Chen, Chengyu Du, Jiangjie Chen, Jinghan Xu, Yikai Zhang, Siyu Yuan, Zulong Chen, Liangyue Li, and Yanghua Xiao. DEEPER insight into your user: Directed persona refinement for dynamic persona modeling. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 24157–24180, Vienna, Austria, July 2025a. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.1177. URL https://aclanthology.org/2025.acl-long.1177/.

Aili Chen, Chengyu Du, Jiangjie Chen, Jinghan Xu, Yikai Zhang, Siyu Yuan, Zulong Chen, Liangyue Li, and Yanghua Xiao. Deeper insight into your user: Directed persona refinement for dynamic persona modeling. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 24157–24180, 2025b. URL https://aclanthology.org/2025.acl-long.1177/.

Jianlyu Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 2318–2335, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.137. URL https://aclanthology.org/2024.findings-acl.137/.

Paul Covington, Jay Adams, and Emre Sargin. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*, RecSys '16, pp. 191–198, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450340359. doi: 10.1145/2959100.2959190. URL https://doi.org/10.1145/2959100.2959190.

Linfeng Du, Ye Yuan, Zichen Zhao, Fuyuan Lyu, Emiliano Penaloza, Xiuying Chen, Zipeng Sun, Jikun Kang, Laurent Charlin, Xue Liu, and Haolun Wu. Optimizing user profiles via contextual bandits for retrieval-augmented LLM personalization. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 31803–31817, San Diego,

California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.1467. URL https://aclanthology.org/2026.acl-long.1467/.

Zhaolin Gao, Joyce Zhou, Yijia Dai, and Thorsten Joachims. Langptune: Optimizing language-based user profiles for recommendation. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, CIKM '25, pp. 707–717, New York, NY, USA, 2025a. Association for Computing Machinery. ISBN 9798400720406. doi: 10.1145/3746252.3761369. URL https://doi.org/10.1145/3746252.3761369.

Zhaolin Gao, Joyce Zhou, Yijia Dai, and Thorsten Joachims. Langptune: Optimizing language-based user profiles for recommendation. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, pp. 707–717, 2025b. URL https://doi.org/10.1145/3746252.3761369.

Gemma. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

Gemma. Gemma 4 technical report, 2026. URL https://arxiv.org/abs/2607.02770.

Qian Guo, Wei Chen, and Huaiyu Wan. Aol4ps: a large-scale data set for personalized search. *Data Intelligence*, 3(4):548–567, 2021.

F. Maxwell Harper and Joseph A. Konstan. The movielens datasets: History and context. *ACM Trans. Interact. Intell. Syst.*, 5(4), December 2015. ISSN 2160-6455. doi: 10.1145/2827872. URL https://doi.org/10.1145/2827872.

Yupeng Hou, Jiacheng Li, Xiangjun Fu, Zhankui He, An Yan, Xiusi Chen, and Julian McAuley. Bridging language and items for retrieval and recommendation: Benchmarking LLMs as semantic encoders. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3251–3265, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.147. URL https://aclanthology.org/2026.acl-long.147/.

Junjie Huang, Jizheng Chen, Jianghao Lin, Jiarui Qin, Ziming Feng, Weinan Zhang, and Yong Yu. A comprehensive survey on retrieval methods in recommender systems. *ACM Transactions on Information Systems*, 44(1):1–43, 2025.

Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446, October 2002. ISSN 1046-8188. doi: 10.1145/582415.582418. URL https://doi.org/10.1145/582415.582418.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM Comput. Surv.*, 55(12), March 2023. ISSN 0360-0300. doi: 10.1145/3571730. URL https://doi.org/10.1145/3571730.

Clark Mingxuan Ju, Yuwei Qiu, Tong Zhao, and Neil Shah. Self-supervised user profile generation for personalization, 2026. URL https://arxiv.org/abs/2606.05336.

Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*, pp. 197–206. IEEE, 2018.

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. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Jixuan Leng, Si Si, Hsiang-Fu Yu, Vinod Raman, and Inderjit S. Dhillon. Groupdpo: Memory efficient group-wise direct preference optimization, 2026. URL https://arxiv.org/abs/2604.15602.

Chao Li, Zhiyuan Liu, Mengmeng Wu, Yuchi Xu, Huan Zhao, Pipei Huang, Guoliang Kang, Qiwei Chen, Wei Li, and Dik Lun Lee. Multi-interest network with dynamic routing for recommendation at tmall. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, CIKM '19, pp. 2615–2623, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450369763. doi: 10.1145/3357384.3357814. URL https://doi.org/10.1145/3357384.3357814.

Lichi Li, Zainul Abi Din, Zhen Tan, Sam London, Tianlong Chen, and Ajay Daptardar. Merrec: A large-scale multipurpose mercari dataset for consumer-to-consumer recommendation systems. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.1*, KDD '25, pp. 2371–2382, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400712456. doi: 10.1145/3690624.3709394. URL https://doi.org/10.1145/3690624.3709394.

Sheng Li and Handong Zhao. A survey on representation learning for user modeling. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, IJCAI'20, 2021. ISBN 9780999241165.

Tianqi Liu, Zhen Qin, Junru Wu, Jiaming Shen, Misha Khalman, Rishabh Joshi, Yao Zhao, Mohammad Saleh, Simon Baumgartner, Jialu Liu, Peter J Liu, and Xuanhui Wang. LiPO: Listwise preference optimization through learning-to-rank. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 2404–2420, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-189-6. doi: 10.18653/v1/2025.naacl-long.121. URL https://aclanthology.org/2025.naacl-long.121/.

Xueguang Ma, Xinyu Zhang, Ronak Pradeep, and Jimmy Lin. Zero-shot listwise document reranking with a large language model, 2023. URL https://arxiv.org/abs/2305.02156.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. On faithfulness and factuality in abstractive summarization. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 1906–1919, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.173. URL https://aclanthology.org/2020.acl-main.173/.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 12076–12100, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.741. URL https://aclanthology.org/2023.emnlp-main.741/.

OpenAI. Openai gpt-5 system card, 2026. URL https://arxiv.org/abs/2601.03267.

Greg Pass, Abdur Chowdhury, and Cayley Torgeson. A picture of search. In *Proceedings of the First International Conference on Scalable Information Systems (InfoScale)*, Hong Kong, June 2006.

Erasmo Purificato, Ludovico Boratto, and Ernesto William De Luca. User modeling and user profiling: A comprehensive survey. *arXiv preprint arXiv:2402.09660*, 2024.

Qwen. Qwen3.6-35b-a3b. https://huggingface.co/Qwen/Qwen3.6-35B-A3B, 2026.

Filip Radlinski, Krisztian Balog, Fernando Diaz, Lucas Dixon, and Ben Wedin. On natural language user profiles for transparent and scrutable recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, pp. 2863–2874, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450387323. doi: 10.1145/3477495.3531873. URL https://doi.org/10.1145/3477495.3531873.

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. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc. URL https://dl.acm.org/doi/10.5555/3666122.3668460.

Xubin Ren, Wei Wei, Lianghao Xia, Lixin Su, Suqi Cheng, Junfeng Wang, Dawei Yin, and Chao Huang. Representation learning with large language models for recommendation. In *Proceedings of the ACM Web Conference 2024*, WWW '24, pp. 3464–3475, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400701719. doi: 10.1145/3589334.3645458. URL https://doi.org/10.1145/3589334.3645458.

Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: Bm25 and beyond. *Foundations and trends® in information retrieval*, 4(1-2):1–174, 2009.

Hinrich Schütze, Christopher D Manning, and Prabhakar Raghavan. *Introduction to information retrieval*, volume 39. Cambridge University Press Cambridge, 2008.

Yunxiao Shi, Wujiang Xu, Zhang Zeqi, Xing Zi, Qiang Wu, and Min Xu. PersonaX: A recommendation agent-oriented user modeling framework for long behavior sequence. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 5764–5787, Vienna, Austria, July 2025a. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.300. URL https://aclanthology.org/2025.findings-acl.300/.

Yunxiao Shi, Wujiang Xu, Zhang Zeqi, Xing Zi, Qiang Wu, and Min Xu. Personax: A recommendation agent-oriented user modeling framework for long behavior sequence. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 5764–5787, 2025b. URL https://aclanthology.org/2025.findings-acl.300/.

Chenkai Sun, Ke Yang, Revanth Gangi Reddy, Yi Fung, Hou Pong Chan, Kevin Small, ChengXiang Zhai, and Heng Ji. Persona-DB: Efficient large language model personalization for response prediction with collaborative data refinement. In Owen Rambow, Leo Wanner, Marianna Apidianaki, Hend Al-Khalifa, Barbara Di Eugenio, and Steven Schockaert (eds.), *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 281–296, Abu Dhabi, UAE, January 2025. Association for Computational Linguistics. URL https://aclanthology.org/2025.coling-main.20/.

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*, CIKM '19, pp. 1441–1450, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450369763. doi: 10.1145/3357384.3357895. URL https://doi.org/10.1145/3357384.3357895.

Katherine Tian, Eric Mitchell, Huaxiu Yao, Christopher D Manning, and Chelsea Finn. Fine-tuning language models for factuality. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=WPZ2yPag4K.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. Multilingual e5 text embeddings: A technical report, 2024. URL https://arxiv.org/abs/2402.05672.

Lu Wang, Di Zhang, Fangkai Yang, Pu Zhao, Jianfeng Liu, Yuefeng Zhan, Hao Sun, Qingwei Lin, Weiwei Deng, Dongmei Zhang, Feng Sun, and Qi Zhang. Lettingo: Explore user profile generation for recommendation system. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, KDD '25, pp. 2985–2995, New York, NY, USA, 2025a. Association for Computing Machinery. ISBN 9798400714542. doi: 10.1145/3711896.3737024. URL https://doi.org/10.1145/3711896.3737024.

Lu Wang, Di Zhang, Fangkai Yang, Pu Zhao, Jianfeng Liu, Yuefeng Zhan, Hao Sun, Qingwei Lin, Weiwei Deng, Dongmei Zhang, et al. Lettingo: Explore user profile generation for recommendation system. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 2985–2995, 2025b. URL https://doi.org/10.1145/3711896.3737024.

Jiaxing Wu, Lin Ning, Luyang Liu, Harrison Lee, Neo Wu, Chao Wang, Sushant Prakash, Shawn O'Banion, Bradley Green, and Jun Xie. Rlpf: reinforcement learning from prediction feedback for user summarization with llms. In *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence and Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence and Fifteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'25/IAAI'25/EAAI'25. AAAI Press, 2025. ISBN 978-1-57735-897-8. doi: 10.1609/aaai.v39i24.34738. URL https://doi.org/10.1609/aaai.v39i24.34738.

Yunjia Xi, Weiwen Liu, Jianghao Lin, Xiaoling Cai, Hong Zhu, Jieming Zhu, Bo Chen, Ruiming Tang, Weinan Zhang, and Yong Yu. Towards open-world recommendation with knowledge augmentation from large language models. In *Proceedings of the 18th ACM Conference on Recommender Systems*, RecSys '24, pp. 12–22, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400705052. doi: 10.1145/3640457.3688104. URL https://doi.org/10.1145/3640457.3688104.

Jiarui Zhang. Guided profile generation improves personalization with large language models. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 4005–4016, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.231. URL https://aclanthology.org/2024.findings-emnlp.231/.

Linhai Zhang, Jialong Wu, Deyu Zhou, and Yulan He. PROPER: A progressive learning framework for personalized large language models with group-level adaptation. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16399–16411, Vienna, Austria, July 2025a. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.800. URL https://aclanthology.org/2025.acl-long.800/.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. Qwen3 embedding: Advancing text embedding and reranking through foundation models, 2025b. URL https://arxiv.org/abs/2506.05176.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

A Appendix

A.1 Motivating the Groundedness Reward

Large Language Models (LLMs) frequently do not produce profiles that faithfully reflect the user's underlying behaviors. We observe three recurring issues, each corresponding to one axis of our reward. First, a profile may group user behaviors that do not share an underlying interest, leaving its evidence cluster incoherent (cohesion). Second, a profile may not adequately capture the common theme across its assigned evidence (alignment). Third, the profile text may overstate or hallucinate claims beyond what the evidence shows (truthfulness). These failures are distinct from one another: a profile can group its evidence well yet mischaracterize what that evidence is, or accurately describe its evidence while also asserting extra, ungrounded claims on top of it. We therefore score each aspect separately rather than as a single judgment. Figure 5 shows one real instance of each, drawn from Qwen3.6-35B-A3B rollouts on `MerRec` users. (exact prompt in Appendix A.8.1)

[Cohesion] "Family-Oriented Household Manager"

Evidence: [0] kids' character T-shirts + [6] WiFi mesh routers + [13] phone cases

Profile text: "... manages procurement for family members, ... balances practical household infrastructure improvements (networking) with lifestyle purchases."

Exploring kids' clothing items, home WiFi routers, and phone cases shares no underlying interest. They are three unrelated behaviors arbitrarily merged into one "household manager" identity.

[Alignment] "Multigenerational Family Caregiver"

Evidence: [0] Kids baseball cleats + [1–10] ten of the women's Nike Pro Shorts

Profile text: "... exhibits a supportive role in family activities, ensuring proper equipment is available for dependents' participation."

Most of this profile's assigned evidence is the exploration of their own athletic shorts; only one unit involves a child at all. The evidence is overwhelmingly personal apparel shopping, but the profile characterizes it wholesale as multigenerational caregiving.

[Truthfulness] "High-End Beauty & Lifestyle Investor"

Evidence: [7] Dyson Corrale hair straightener

Profile text: " Invests in premium, technology-forward personal care tools ... Aligns personal grooming habits with a preference for quality and efficacy. "

A stable "investor" identity and recurring grooming "habits" (plural) are asserted from a single evidence.

Figure 5: Three independent groundedness failures in profiles produced from Qwen3.6-35B-A3B, drawn from real MerRec logs. **Cohesion**: unrelated evidence merged into one profile. **Alignment**: a profile mischaracterizes the evidence behaviors. **Truthfulness**: a stable identity/habit is asserted from a single, thin observation. These motivate directly optimizing for groundedness.

These patterns are not limited to open-weight models. Running the identical prompt with two strong proprietary models, Claude Sonnet 5 (Anthropic, 2026) and GPT-5.1 (OpenAI, 2026), produces the same kinds of issues, suggesting that they are not resolved by model capability alone.

Truthfulness. Claude Sonnet 5 produces a profile titled "Female pop artist superfan across formats" whose text mentions Sabrina Carpenter, Demi Lovato, and figurines, none of which appear among its 8 assigned evidence units. In addition, GPT-5.1 builds a profile from a single unit about generator specifications (electric start and inverter), titled "Pragmatic Household Planner Preparing for Contingencies," and describes the user as showing "concern for power security or preparedness (e.g., outages, events)." The unit contains no mention of outages or emergencies, so this user profile text is not supported by the evidence.

Cohesion. GPT-5.1 merges behavioral units about living-room couch, Lululemon athletic apparel, camera bag, and Prada sunglasses into a single profile, "Pragmatic Home-and-Appearance Upgrader," describing it as treating "home comfort and personal appearance as parallel projects to refine over time." The couch shares no evident underlying interest with the apparel or accessories, yet all four are grouped together.

Alignment. On a user's behavioral units where 10 of 11 are repeated behaviors of the same women's athletic clothes and only one unit concerns a child's baseball cleats, GPT-5.1 assigns all 11 units to a single profile, "Caregiver balancing own athletic needs with children's sports gear," describing it as reflecting "a household sports routine" with "integrated planning around shared sports or activity schedules." The profile characterizes evidence that is almost entirely the user's own apparel shopping as being about coordinated family activity.

We find that these recurring patterns reflect general properties of profile generation without a groundedness objective, rather than artifacts specific to one backbone.

A.2 Dataset Details

Platform and domain. `Srv.` is collected from a nationwide online platform operating both a search service and a recommendation feed, whose documents span multiple domains, including web, news, entertainment, sports, and user-generated content. `MerRec` (Li et al., 2025) is collected from Mercari, a consumer-to-consumer online marketplace. `AOL4PS` (Guo et al., 2021) is built from the AOL web search engine query log (Pass et al., 2006).

Item metadata. In `Srv.`, each document is represented by its title. In `MerRec`, each product is represented by its title, hierarchical categories, brand, and price. In `AOL4PS`, each website is represented by the crawled webpage text provided with the dataset.

User sampling. For all datasets, we randomly sample 10,000 users from those active on at least 14 days within the 8-week period, ensuring each sampled user has enough behavioral history. For `Srv.`, which spans multiple domains, we additionally stratify users to cover diverse usage patterns. Using the 180 days of logs before the 8-week period, we count a day as active in a domain if the user clicks at least three documents in that domain, and label each user per domain by the fraction of these 180 days that are active (heavy: $\geq 80\%$, medium: $\geq 50\%$, light: otherwise). After excluding users who are light in every domain, we stratify the rest by their set of (domain, activity level) pairs and sample randomly within each stratum. The sampled users are split into training, validation, and test sets at a 7:1:2 ratio as described in Section 4.1.

A.2.1 L2 ranking slate construction.

Each L2 ranking slate consists of one positive item and 19 hard negatives. To make the ranking task non-trivial, hard negatives are selected to be plausible alternatives rather than items that can be easily distinguished from the positive. For `Srv.`, they are sampled from recommendation impressions shown to the same user, whereas for `MerRec`, which does not provide impression logs, they are constructed using the sampling procedure described below.

`MerRec.` Using the positive item itself as the query, we retrieve the top 50 candidates from the full item pool via sparse retrieval (BM25; Robertson & Zaragoza, 2009) and the top 50 via dense retrieval (BGE-M3; Chen et al., 2024), merge the two candidate sets, and deduplicate. We then randomly sample 19 items from this deduplicated pool as hard negatives and insert the positive item at a randomly assigned position within the resulting 20-item slate. Retrieving negatives using the positive item as the query is intended to surface hard negatives that plausibly appeal to the user, since they resemble an item the user is already known to like, without having actually been interacted with. Randomizing the positive's position controls for positional bias in the ranking function. This slate is constructed once per positive item and reused identically across all methods' evaluations.

`Srv.` Because the service log provides recommendation impression logs, we construct the hard negative pool separately for each (user, target window) pair. The pool contains all documents shown to the user in the recommendation feed during the target window that were not clicked. Thus, the

pool consists of items the user was exposed to but did not interact with. For each user's positive item, we randomly sample 19 negatives from this pool to form the ranking slate.

A.3 Behavioral Units Aggregation

Raw interaction logs are noisy and redundant at the event level. A single interest surfaces as many near-duplicate events scattered across a day. As a preprocessing step, we therefore consolidate each user's log into behavioral units. Each day of a user's events is passed in full to an off-the-shelf instruction-tuned model `gemma-3-12b-it` (Gemma, 2025), prompted to merge temporally or topically related events into a single unit and keep distinct intents separate. A day therefore yields several behavioral units, and each unit summarizes a contiguous or scattered subset of that day's events. Consolidation is applied once over the full log period, but units are used only where a window serves as input L_u^{past}; when a window serves as the target L_u^{future}, the raw events are used directly, since downstream evaluation ranks the individual items a user interacted with. The same procedure is applied to every dataset, and its resulting units are fixed before any training. Figure 6 shows one such unit, drawn from a single day of `MerRec` logs. Six interactions with different Casio watch models, mixing views and likes, are merged into a behavioral unit, while the surrounding events from that day, elided here, form units of their own. Figure 7 shows the same procedure applied to a search log from the `AOL4PS` dataset. A search query for wholesale silver jewelry, together with four clicks on distinct wholesale-jewelry sites, is merged into a single behavioral unit, while the surrounding events from that day, elided here, form units of their own.

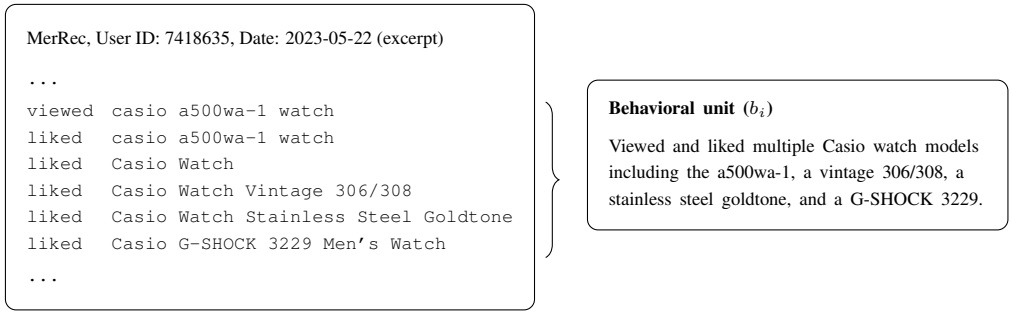

Figure 6: Consolidation of raw interaction events into a behavioral unit from the `MerRec` dataset. Interactions with several Casio watch models are merged into one behavioral unit.

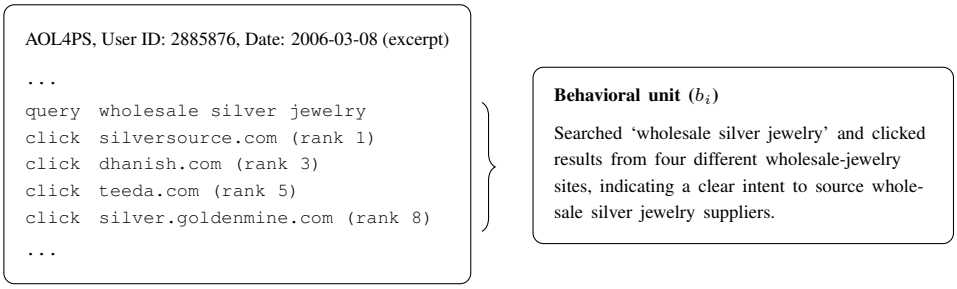

Figure 7: Consolidation of raw interaction events into a behavioral unit from the `AOL4PS` dataset. A repeated query with several distinct clicked results is merged into one behavioral unit.

A.4 Alignment & Truthfulness Judge

Setup. As defined in Section 3, each profile $p_i = (E_i, y_i)$ is scored by two LLM-based judges. J_{align} takes (y_i, E_i) and counts the fraction of evidence units in E_i that directly support y_i, following the prompt in Figure 10. J_{truth} separately scores the profile title and each description for claims

not grounded in E_i, penalizing exaggerated language unless directly confirmed by the evidence, following the prompt in Figure 9 (both in Appendix A.8.2). Both judges output a calibrated score in $[0, 1]$. Our framework is self-contained: the same backbone model both initializes the policy and serves as the two judges. We use QWEN3.6-35B-A3B for our main experiments and GEMMA-4-26B-A4B-IT for the backbone-generalization experiments in Appendix A.5.

Judge reliability. To check whether the choice of judge model drives the resulting scores, we sample 500 individual profiles from training rollouts on `MerRec` and additionally score each with CLAUDE SONNET 5 (Anthropic, 2026) and GPT-5.1 (OpenAI, 2026), using the identical prompts from Figures 9–10. Table 6 reports the resulting agreement between three judges. Scores from the two external judges differ from QWEN's by only 0.06–0.11 points on average (on the $[0, 1]$ scale), and fall within 0.2 of QWEN's score in 85–89% of profiles, for both alignment and truthfulness. CLAUDE and GPT agree with each other as well (0.04–0.11 MAD, 89–94% within 0.2). Overall, this suggests the groundedness scores are reliable across judge models.

Table 6: Inter-judge reliability on 500 randomly sampled profiles. For each pair of judges, we report the mean absolute difference (MAD) between their scores on the same profile, and the percentage of profiles on which the two scores differ by at most 0.2.

	QWEN VS. CLAUDE MAD (% within 0.2)	QWEN VS. GPT MAD (% within 0.2)	CLAUDE VS. GPT MAD (% within 0.2)
Alignment (J_{align})	0.063 (88%)	0.056 (89%)	0.045 (94%)
Truthfulness (J_{truth})	0.104 (88%)	0.113 (85%)	0.109 (89%)

Qualitative examples. Table 7 shows two representative profiles from this sample. In the first, all three judges independently assign near-perfect scores to a profile whose evidence cluster is uniformly and specifically about *WWE Ultimate Edition action figures*. In the second, all three judges independently assign scores near 0: the profile claims the user *browses both men's and women's clothing*, but every evidence unit in its cluster is in fact a repeated view of the same women's hoodie, with no mention of men's apparel at all.

Table 7: Two sampled profiles $p_i = (E_i, y_i)$ and the scores assigned by all three judges. Q: Qwen, C: Claude, G: GPT.

	Profile + Evidence	**align / truth**
Well-grounded	*WWE Ultimate Edition Collectors Intent*: "Focuses exclusively on the specific 'Ultimate Edition' line…targets specific iconic figures…" + Evidence: 5 units, all repeated behaviors of viewing WWE Ultimate Edition figures.	Q: 1.00 / 0.75 C: 1.00 / 0.75 G: 1.00 / 0.83
Ungrounded	*Dual-Gender Household Shopper*: "Navigates distinct clothing categories for both men and women…switches gender filters explicitly…" + Evidence: 3 units, all repeated behaviors of viewing women's hoodies.	Q: 0.00 / 0.15 C: 0.00 / 0.00 G: 0.00 / 0.00

A.5 BACKBONE GENERALIZATION

To examine whether the observed transfer generalizes beyond a single model family, we replace the base model with GEMMA-4-26B-A4B-IT on `MerRec` and `AOL4PS`, keeping the downstream evaluation setup unchanged (Table 8). Groundedness training again improves over the untrained GEMMA model across every setting, with statistically significant gains throughout ($p < 0.001$). As with QWEN3.6-35B-A3B, the improvement is largest on L1 retrieval, where OURS improves over BASE by 84.6% and 80.9% with E5 and BGE, respectively. Gains on L2 ranking and personalized search are more modest but consistent, ranging from 1.0% to 2.5% across rankers. These results suggest that groundedness training generalizes across model families.

Table 8: Test-set performance on MerRec and AOL4PS with GEMMA-4-26B-A4B-IT as the backbone (BASE) against groundedness training (+ OURS). Metrics follow Table 4: Recall@100 for retrieval (L1) and NDCG for recommendation ranking (L2) and personalized search (PS). ‡ denotes a statistically significant improvement over BASE (paired t-test) at $p < 0.001$.

	MerRec					AOL4PS		
	L1		**L2**			**PS**		
Method	E5	BGE	E5	Q3-R	LRL	E5	Q3-R	LRL
BASE	0.0123	0.0136	0.3579	0.3928	0.4504	0.5193	0.5599	0.6315
+ OURS	0.0227‡	0.0246‡	0.3637‡	0.3968‡	0.4544‡	0.5310‡	0.5739‡	0.6433‡

A.6 IMPLEMENTATION DETAILS

This section describes the training pipeline shared by all methods (Common), followed by the reward-specific details for each (OURS, RLPF, LETTINGO).

Common All methods share a same training pipeline and differ only in the reward function used to score each rollout. The same $n = 4$ candidate rollouts, generated once per (user, input window) using the profile-generation prompt (Figure 8, Appendix A.8.1), only the reward function attached to them differs. The training rollouts are sampled at a high temperature (temperature $= 1.0$) to encourage diversity across the group, since the groupwise DPO objective relies on meaningfully different candidates within a group to produce a useful training signal. The backbone is loaded in bf16 and trained with LoRA ($r = 16$, $\alpha = 32$, dropout $= 0.05$) added as a trainable adapter, alongside a frozen copy of the same LoRA adapter that serves as the reference policy for groupwise DPO; only the trainable adapter's parameters receive gradients. We optimize with AdamW (lr $=$ 1e$-$5, $\beta = (0.9, 0.95)$, $\epsilon =$ 1e$-$8, weight decay $= 0$) under a cosine schedule with a 5% linear warmup, gradient-clipped at 0.5 norm, for 5 epochs over the training split. For all three methods, the reward for each rollout is computed using the same base backbone model as the one being trained and called near-deterministically (temperature $= 0.1$) for consistent scoring.

Ours The cohesion penalty weight is fixed at $\lambda = 0.5$. The cluster-size regularizer uses $\tau_{\text{size}}^{\min} = 3$ and $\tau_{\text{size}}^{\max} = 20$ for MerRec and AOL4PS. For the denser Srv. (Section 4.1), we increase $\tau_{\text{size}}^{\max}$ to 60, so that the penalty begins when a single profile covers roughly one third of an input window. The three groundedness components (alignment, truthfulness, and cohesion) are equally weighted, and their aggregate is combined with the size regularizer using a 9:1 ratio ($\alpha_1 = 0.9$, $\alpha_2 = 0.1$), keeping the regularizer as a relatively weak auxiliary constraint. At the profile-set level, the mean per-profile reward is similarly combined with the coverage regularizer using a 9:1 ratio ($\alpha_3 = 0.9$, $\alpha_4 = 0.1$). These reward weights are fixed across datasets rather than tuned separately for each dataset. The alignment and truthfulness judges use the prompts in Figures 10 and 9 (Appendix A.8.2).

RLPF For each item the user actually interacted with in the target window, we construct a four-way multiple-choice question. The true item is placed at a randomly chosen position among four candidates, with the remaining three drawn uniformly at random from the full item catalog. Given the generated profile, the judge model is prompted to select the index of the item the user will interact with next; the reward for that item is 1 if the judge's selection matches the true position and 0 otherwise. When a user has multiple target-window items, we average this binary score to obtain the rollout's final reward. We adapt the next-activity prediction prompt from Wu et al. (2025):

```
Here is a summary of a user's preferences based on their
past activity (between <summary> and </summary>).
<summary>{profile}</summary>

Based on this information, which item will this user
interact with next?
Choose one from the following items:
0. {item0}
1. {item1}
2. {item2}
```

```
3. {item3}
Structure the answer with the item letter only without extra
words.
```

LettinGo For each item the user actually interacted with in the target window, we construct two candidates: the true item and one random negative drawn from the full item catalog. Given the generated profile, the judge model is prompted to classify the user's likely sentiment toward each candidate as *like*, *neutral*, or *dislike*; we treat *like* as positive and both other labels as negative. The reward for a rollout is the classification accuracy across all such candidate pairs for that user (i.e., whether the true item is classified *like* and the random negative is not), averaged over all target-window items. We adapt the sentiment classification prompt from Wang et al. (2025b):

```
Given a user profile which depict the user's interest about
items, your task is helping me predict a user's possible
sentiment about a target item. The sentiment has three
categories: like, neutral, and dislike. Remember, your
output should only contain one word (like, neutral or
dislike, in lowercase) that represent user sentiment you
predict, without any additional content.
USER PROFILE: {profile}.
The candidate item is: {item}.
```

A.7 Qualitative Case Studies

This section provides five representative case studies from the MerRec test set, each isolating one pattern from Section 5. Case Study A shows OURS declining to build a profile from a single ambiguous unit. Case Study B shows RLPF splitting a thinly-supported niche interest into its own profile. Case Studies C and D show, respectively, OURS and LETTINGO describing the same evidence as BASE more specifically. Case Study E shows that OURS+RLPF exhibits traits of both rewards: while it follows OURS in declining to build or split off profiles from a single unit (Case Studies A and B), it retains RLPF's split of distinct interests.

Case Study A: OURS declining to invent a profile from thin evidence.

> **User 56253432** (2023-05-29 to 2023-06-11, MerRec) has 42 behavioral units in total. BASE organizes 41 of them into three well-supported profiles: a 28-unit vintage-glass-collecting profile (three of its units, for example, are:
>
> **[0]** "Vintage Whitehall glassware…repeatedly views and likes…tumblers, glasses, and sets."
>
> **[1]** "Pink Depression Glass…views and likes several items…salt and pepper shakers, candle holders, and bowls."
>
> **[2]** "Federal Glassware…views multiple items…Patrician Spoke Dinner Plates, Cubist Glasses, and Cabbage Rose pie plates."
>
>), an 8-unit athletic-apparel profile, and a 4-unit vintage-kitchenware profile. The one unit left over does not fit any of these:
>
> **[7]** "Baby carrier blankets…the user viewed a baby carrier blanket, indicating a potential interest in purchasing items for a baby."
>
> The question is what each method does with this one leftover unit.
>
> → **BASE** builds the three well-supported profiles above, plus a 4th profile from unit [7] alone, *The Anticipatory Parent*: "…suggesting a *life stage transition* or specific family responsibility."
>
> → **RLPF** does the same, building *Nursery & Baby Product Explorer* from unit [7] alone: "Indicates a *potential life stage change* or interest in preparing for a child."
>
> → **LETTINGO** does the same, building *Nursery and Family Preparation* from unit [7] alone.

> → **OURS** builds the same three well-supported profiles, but no profile from unit [7]: it is simply left unassigned.
>
> → **OURS+RLPF** makes the same choice as OURS: no profile is built from unit [7].

Where BASE, RLPF, and LETTINGO all extend a single ambiguous unit into a profile-level claim about the user's life stage, OURS treats it as insufficient grounds for a profile and moves on.

Case Study B: RLPF splitting a niche interest into its own profile.

> **User 39539469** (2023-06-05 to 2023-06-18, `MerRec`), 11 behavioral units, including music-related interests:
>
> **[0]** King Gizzard and the Lizard Wizard vinyl records
>
> **[1,2,3]** three King Gizzard tour/concert posters
>
> **[4,10]** King Gizzard pin collections (viewed, liked, made an offer)
>
> **[8]** Metallica Black Album cassette singles, from a listing for "…Cassette Singles Lot Of 5"
>
> → **BASE** merges all 7 units (King Gizzard [0,1,2,3,4,10] + Metallica [8]) into one profile, *Dedicated Music Memorabilia Collector*.
>
> → **LETTINGO** makes the same merge, keeping the same profile name and the same 7 units.
>
> → **OURS** makes the same merge, with a more delineated profile title: *Dedicated Music & Band Memorabilia Collector*.
>
> → **OURS+RLPF** makes the same merge as well, and additionally consolidates other small clusters elsewhere in the log rather than splitting them further.
>
> → **RLPF** instead splits unit [8] off into its own profile, *Alternative Music Format Archivist*, describing it as suggesting "a deep dive into music history" and a preference for physical formats "over digital streaming."

Where BASE, LETTINGO, OURS, and OURS+RLPF all fold the Metallica listing into the same broader collector profile, RLPF alone peels it off into a separate profile, attaching claims a single behavioral unit cannot support. This is consistent with a broader pattern: RLPF tends to favor extracting an additional distinguishing trait, even when the evidence for that trait is thin.

Case Study C: OURS inducing more specific profile on identical evidence.

> **User 32528103** (`MerRec`), 14 behavioral units, selected identically by BASE and OURS: repeated views of Blink outdoor/indoor cameras, a video doorbell, mounting brackets, and sync modules, several explicitly listed as NIB (new in box).
>
> → **BASE** induces *Systematic Home Security Integrator*: "Conducts deep, iterative research on specific hardware ecosystems (Blink) to build a comprehensive surveillance network."
>
> → **OURS** induces *Blink-Centric Home Security System Builder*: "Heavily focused on the Blink ecosystem, repeatedly viewing *cameras in Outdoor, Mini, Indoor, doorbells*, and specific accessories like mounts and sync modules …checking specific conditions like *NIB*."

On the same evidence, OURS describes the user's interest more specifically than BASE.

Case Study D: LETTINGO naming the same evidence more precisely than BASE.

> **User 33769279** (`MerRec`), 8 behavioral units, selected identically by BASE and LETTINGO: repeated views of AirPods and AirPods Pro, searches for Bluetooth headphones compatible with iPhone, and one listing for AirPods Pro parts only.
>
> → **BASE** induces *The Tech-Savvy Ecosystem Optimizer*: "Seeks seamless integration between hardware and software, specifically prioritizing device compatibility (iPhone) and brand synergy (Apple) … values reliability and ecosystem lock-in."

> → **LETTINGO** induces *Apple Ecosystem Audio Upgrader*: "Actively searches for Bluetooth headphones specifically compatible with iPhone ... considers both premium options (AirPods Pro) and potential DIY/repair aspects."

BASE describes the user as a general *Tech-Savvy Ecosystem Optimizer*, while LETTINGO names *Apple Ecosystem Audio Upgrader*, committing to the actual product category (audio) and the actual behavior (upgrading) rather than staying at an abstract level.

Case Study E: OURS+RLPF preserving a well-supported niche split.

> **User 48535431** (`MerRec`), 6 behavioral units spanning two related-but-distinct interests: home-office ergonomics (desk lamps, laptop stands) and musical-instrument accessories (Yamaha keyboard stands).
>
> → **BASE** merges all 6 units into one profile, *The Ergonomic Home Office Optimizer*.
>
> → **OURS** also merges all 6 units into one profile, *Tech-Savvy Home Office Setup Seeker*.
>
> → **RLPF** splits the same 6 units into *Home Office Ergonomics Optimizer* (4 units: desk lamps, laptop stands) and *Musical Instrument Accessory Planner* (2 units: Yamaha keyboard stands).
>
> → **OURS+RLPF** preserves this same split: *Home Office Ergonomics and Setup Optimizer* (4 units) and *Musical Instrument Accessory Seeker* (2 units).

Unlike Case Study B, OURS+RLPF keeps this split rather than reverting to OURS's consolidation. Read together with Case Study B, this suggests RLPF's inclination to split off niche interests is a mixed signal, sometimes overreaching beyond what the evidence supports, sometimes capturing a genuine distinction that groundedness alone would merge away.

A.8 PROMPTS

This section provides the full prompts used in our framework. Section A.8.1 shows the main prompt used to generate user profiles, and Section A.8.2 shows the two judge prompts used to score alignment and truthfulness for the groundedness reward.

A.8.1 MAIN PROMPT

Figure 8 shows the prompt used to generate user profiles from behavioral units. The prompt instructs to identify recurring patterns, and to organize them into profiles supported by evidence units.

```
# User Profiles Generation Prompt
```

You are an analytical assistant that induces user profiles from user behavioral units.

Core Principles:
- A profile is not a simple topic or interest classification.
- A profile must reveal the following elements together:
  1) Repeated behavioral patterns
  2) Life context or rhythm
  3) Personal tendencies or values
- Even if topics differ, group them into one profile if the exploration style clearly overlaps.
  - However, only group different topics together when the context clearly overlaps.
  - Avoid overly abstract or excessively broad profiles.
- A single unit (behavior_id) may belong to multiple profiles.
  - However, allow overlap only when the context clearly intersects.

Input:
- A user's search or recommendation service behaviors sorted in chronological order.

Task:
1) Identify recurring patterns in how the user explores information and structures their daily life.

2) Based on these patterns, derive profiles that reflect what they value.
3) Each profile must include:
- A concise noun phrase or descriptive phrase that conveys the tendency (title)
- 2–3 behavioral signatures that characterize the tendency (description)
4) Avoid exaggerated expressions.
5) Explicitly indicate which behavioral units (behavior_id) support each profile.

The output must strictly follow the JSON format below.

```
{
  "profiles": [
    {
      "profile_id": "P1",
      "title": "<One concise phrase describing the behavioral pattern.>",
      "description": ["<key behavior 1>",
                      "<key decision criterion>",
                      "<typical constraints>"],
      "evidence_ids": [<behavior_id>, <behavior_id>, <behavior_id>, ...]
    },
    ...
  ]
}
```

User Behavioral Units:
{{ B_u^{past} }}

Figure 8: Prompt used to generate user profiles from user behavioral units.

A.8.2 JUDGE PROMPTS

Figures 10 and 9 show the prompts used for LLM-based scoring of profile groundedness, corresponding to the alignment and truthfulness, respectively.

```
# Truthfulness Judge Prompt
```

You are a highly strict evaluator of exaggeration or unsupported claims.

Objective:
- For a profile title and for each description, determine whether they are supported by the provided evidence.
- Any inference, speculation, or generalization not explicitly grounded in the evidence should be penalized.

Important Rules:
- Strong expressions such as "repeatedly," or "always," are considered exaggerated unless repeated patterns are clearly confirmed in the evidence.
- If there is little to no relevant evidence, assign a low score (close to 0).

Scoring Guidelines (0.0–1.0):
- 1.0: Clearly and repeatedly supported by evidence; no exaggeration.
- 0.7: Mostly supported, but some wording is slightly strong (minor exaggeration).
- 0.3: Partially supported. Some core claims lack evidence or are generalized.
- 0.0: Little to no supporting evidence. Primarily speculative or exaggerated.

Output must strictly follow the JSON format below:

```
{
  "title": {
    "score": <0.0–1.0>, "overclaim_phrases": ["<short phrase judged as exaggerated>", ...]
  },
  "descriptions": [
    {
      "id": 1,
      "score": <0.0–1.0>, "overclaim_phrases": ["<short phrase judged as exaggerated>", ...]
    },
    ...
  ]
```

}

Input:
{{ p_i }}

Figure 9: Prompt used for evaluating profile truthfulness. The judge assesses whether profile descriptions are supported by the evidence and penalizes exaggerated or unsupported claims.

Alignment Judge Prompt

You are a strict evaluator assessing the alignment between a profile and its supporting behaviors. Below, one profile description and multiple evidence behaviors are provided.

Objective:
- Determine whether each evidence meaningfully aligns with the core characteristics of the profile.
- Count the number of evidence items that clearly align with the profile.

Evaluation Criteria:
- If an evidence item serves as direct support for the profile, count it as aligned.
- If it is weakly related or reflects a different tendency, do not count it as aligned.

Score Calculation:
- score = (number of evidence items aligned with the profile) / (total number of evidence items)

Output must strictly follow the JSON format below:
{
"score": <0.0–1.0>,
"aligned_evidence_ids": [<int>, ...],
"non_aligned_evidence_ids": [<int>, ...]
}

Input:
{{ p_i }}

Figure 10: Prompt used for evaluating profile–evidence alignment. The judge determines which behavioral evidence support the profile and computes an alignment score based on the proportion of aligned evidence.