REASONING RERANKING WITH SEMANTIC IDS AS THE SOLE ITEM REPRESENTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Reasoning rerankers write a rationale before ranking a candidate slate, and existing systems read item metadata to do so, even though generative recommenders increasingly represent items as compact Semantic IDs (SIDs). Ranking from SIDs alone has two appeals: SID prompts are far shorter than the matched item text, and although item codes are fixed from content, the backbone learns their token embeddings in part from user–item interaction sequences, so the tokens can carry behavioral signal that item text lacks. That training targets next-item generation, and whether it supports ranking a supplied slate from SIDs alone is untested. We build RERANKID, a controlled testbed in which models reason and rank from a history and slate shown only as SIDs, varying item rendering, history length, and slate size one at a time. On this testbed, such models often copy the input order instead of reranking, and readable metadata does not repair the failure. We compare supervised fine-tuning, rejection-sampled self-training, on-policy distillation, and verifier-based reinforcement learning as post-training signals and find that they elicit the missing behavior. The strongest policy, an 8B model trained with a graded ranking reward, gives the best SID-only ranking and matches strong text-reading LLMs whose prompts are about $20\times$ longer. It ranks above random with no history at all, where a caption-reading reference does not, and it keeps its gains at long histories, where text prompts pass 50k tokens and the text reference stops improving. A paired rationale audit separates ranking from description: the policy argues its trade-offs more sharply than the text reference, but its claims about specific items are often wrong. SIDs are enough to learn the ranking; describing the items from SIDs alone is an open grounding problem.

1 INTRODUCTION

Modern recommenders choose what to show through a cascade of stages, and the final stage reranks a given slate of candidates for the user. This stage depends on detailed comparison against the user's history and among the candidates, and reasoning rerankers make this comparison explicit: before the final ranking, they verbalize a rationale a reader can check (Sun et al., 2023; Pradeep et al., 2023; Gangi Reddy et al., 2024; Hou et al., 2024). In parallel, generative recommenders increasingly represent items as compact Semantic IDs (SIDs) (Rajput et al., 2023; Hua et al., 2023; Zhou et al., 2026). Two properties of these identifiers suit reranking. First, they are short. A text rendering of a long history and slate runs to tens of thousands of tokens, which is costly and is where LLM ranking weakens (Liu et al., 2024; Hou et al., 2024); the matched SID rendering stays in the hundreds. Second, their token embeddings are learned in part from behavior. In OpenOneRec, each item's code is assigned by quantizing a text embedding of the item, and the backbone then learns the embedding of each code token in a pretraining stage that aligns it with text and trains it on user behavior sequences, the stage OpenOneRec credits with encoding collaborative signal (Zhou et al., 2026). The tokens can therefore carry preference signal that titles and captions do not contain.

Recent SID reasoners align SID tokens with text in one backbone and reason in language, but they generate the next item from a history rather than rank a given slate (Liu et al., 2026; He et al., 2026; Zhou et al., 2026). Listwise reasoning rerankers order an externally supplied slate, but they replace or augment the identifiers with readable metadata (Liang et al., 2026; Zhang et al., 2026). Neither line of work shows whether a model can rank a given slate of candidates using only what its SID tokens learned in pretraining.

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.

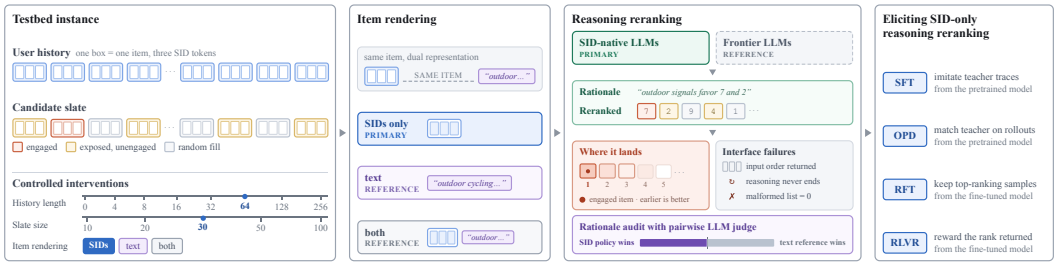

Figure 1: Overview of RERANKID. Models read an SID-only history and candidate slate, produce a rationale, and return a ranked index list; a paired audit compares the emitted rationales. SFT and OPD start from the pretrained model, RFT and RLVR continue from the SFT checkpoint, and each is compared with its own initializer.

We study the missing intersection: *reason-then-rerank from SIDs alone*. At inference time, both the user history and every candidate are rendered only as SIDs; natural-language task instructions are allowed, but no item text or derived user profile is shown. RERANKID is the frozen testbed we build for this controlled study (Figure 1). On the testbed, a model produces a rationale followed by a ranking of candidate indices. The ranking refers to candidates only by index, so the same users and slates can also be rendered as text alone or as both SIDs and text without changing the task. The user-disjoint testbed changes the item rendering, the history length, and the slate size along one axis at a time, and scores ranking quality alongside interface failures such as reasoning loops and copying the input order. We treat the rationale as what a reasoning reranker shows its reader and audit its content directly, without assuming that it causes the ranking.

We first diagnose the existing models before any post-training. The dominant failure is the interface: at the anchor setting (a 64-item history and a 30-candidate slate), both models trail the text-view references and frequently return the candidate order unchanged, and readable captions change how they fail rather than how well they rank. The SID prompt is roughly $20\times$ shorter than its text counterpart and nearly $28\times$ shorter at the longest history, yet longer histories bring the existing models no sustained benefit.

We then ask whether common post-training signals can elicit reasoning reranking from an SID-pretrained model. We define matched 1.7B and 8B recipes for teacher-supervised SFT, verifier-filtered RFT (Zelikman et al., 2022), OPD on student-generated behavior (Agarwal et al., 2024), and RLVR with a ranking verifier (Shao et al., 2024). SFT alone removes the identity and loop failures and yields clear ranking gains at both scales. At 8B, RLVR then improves on SFT with a resolved paired gain (its 95% user-paired bootstrap interval excludes zero) and gives the strongest SID-only policy, on par with text-reading LLMs that see prompts about $20\times$ longer; at 1.7B it matches SFT. The graded ranking reward also beats a binary top-one variant in the direct paired contrast at 8B. The elicited policy ranks above random with no history at all, where a caption-reading reference does not, and it keeps its gains at the longest histories, where the text reference stops improving. A paired rationale audit then separates ranking from description: the strongest SID policy argues its trade-offs more sharply than the text reference, yet it misstates specific item content it has never seen as text. Taken together, the results describe what the SID tokens give a reasoning policy: item-level preference signal and usable history evidence, which suffice to order a slate, but not the item-level detail needed to describe it.

Our contributions are threefold:

- We define pure-SID reasoning reranking and conduct a controlled study of it, built on a released frozen testbed[1] with paired item renderings, one-axis history and slate interventions, and interface-failure diagnostics.
- We diagnose the existing models before any post-training: they neither reliably handle the listwise interface nor benefit from longer histories, and readable captions do not repair either failure.

[1] Anonymized code: https://anonymous.4open.science/r/rerankid-code-8D2B.

- We compare four common post-training signals under one protocol and find that the elicited policy uses signal a text reader does not: it ranks above random without history and keeps its gains at long histories where a text reference stops improving. A paired rationale audit shows that this ranking ability does not come with reliable item descriptions.

2 Related Work

Semantic IDs and reasoning over them. Generative recommenders index items as compact sequences of discrete Semantic ID tokens and generate the next item directly in this token space (Rajput et al., 2023; Hua et al., 2023). OpenOneRec releases 1.7B and 8B models built on one SID tokenizer, together with training data and pipeline, which makes controlled downstream studies possible (Zhou et al., 2026). One line examines the representation itself: attaching SID tokens to a language model does not by itself make their semantics usable (Chen et al., 2026), and the tokens' capacity and decoding structure limit what they express (Hou et al., 2026). Another aligns SID tokens with text in one backbone and adds verbalized or staged reasoning before item generation (Liu et al., 2026; Team et al., 2026; He et al., 2026). Because the main decision in these systems remains which catalog item to generate, reasoning quality and decoding constraints cannot be separated. We instead fix one existing model family, remove the generation step, and test whether its pretrained SID tokens alone support reasoning over a supplied slate.

Reranking with large language models. Listwise rerankers prompt or train LLMs to order retrieved documents (Sun et al., 2023; Pradeep et al., 2023; Gangi Reddy et al., 2024), and recent systems devote explicit test-time reasoning to this step (Weller et al., 2025; Yang et al., 2025), which REARANK trains with reinforcement learning from a listwise NDCG reward (Zhang et al., 2025). In recommendation, LLMs can rerank candidate items without task-specific training, but their accuracy is sensitive to history length, candidate-set size, and presentation order (Hou et al., 2024; Bito et al., 2026); ALRO trains the ranking directly with a differentiable listwise loss and a permutation-consistency term against position bias (Chao et al., 2024). The GR2 family is closest to our setting: a model generates a rationale and a ranking for a supplied slate (Liang et al., 2026), with industrial and diffusion variants training the same behavior at scale (Li et al., 2026; Zhang et al., 2026). Related designs reason pointwise before scoring each candidate (Zheng et al., 2026a) or rerank generatively without a verbalized rationale (Jia et al., 2026). No system in this line verbalizes its reasoning from item identifiers alone: wherever a rationale appears, readable item text is in the prompt. Such systems cannot show whether identifier tokens alone carry preference signal, or whether a rationale stays accurate once item text is withheld. ReRankID withholds that text. SIDs are the sole item representation, and matched text views turn metadata access into a controlled variable, which makes both questions measurable.

Post-training for reasoning rerankers. Standard recipes for eliciting reasoning include rejection self-training (RFT) (Zelikman et al., 2022), on-policy distillation (OPD) (Agarwal et al., 2024), and reinforcement learning with verifiable rewards (RLVR) (Shao et al., 2024). Recommendation adaptations distill rationales into smaller recommenders (Wang et al., 2024; Xu et al., 2025) and shape ranking behavior with rank-aware rewards (Zheng et al., 2026b; You et al., 2025); Rec-R1 optimizes the LLM with GRPO against the downstream ranking metric itself, including product reranking, and compares this with SFT, rejection sampling, and DPO on text-based product search (Lin et al., 2025), while the industrial GR2 pipelines mix several recipes (Li et al., 2026). We run SFT, RFT, OPD, and RLVR as matched arms on one pure-SID task, where items are never shown as text, so that each result, positive or null, can be judged against its own supervision budget.

3 The ReRankID Testbed

We want a controlled study of two questions: whether the existing models can rerank candidates from SID tokens alone, and which post-training signals elicit that behavior. No existing resource can answer them. The study needs the same users and slates to be rankable from SIDs, from item text, and from both, so that metadata access is a variable rather than a confound. It needs every other factor frozen while one intervention varies. And it needs interface failures separated from ranking

	Input	NDCG@5	NDCG@10	Recall@5	Recall@10	MRR	Identity%	Loop%	Input tokens
Random	–	.105	.157	.190	.350	.099	–	–	–
Caption	Text	.184	.248	.303	.503	.171	–	–	–
Claude-Sonnet-5	Text	.210	.282	.315	.540	.205	0.0	0.0	16,722
Gemini-3.6-Flash	Text	.247	.310	.378	.570	.231	0.0	0.0	16,722
GPT-5.6-Terra	Text	.256	.322	.398	.603	.237	0.0	0.0	16,722
OpenOneRec-1.7B-pro	SID	.114	.167	.195	.365	.109	42.3	16.5	845
OpenOneRec-1.7B-pro	Text	.115	.165	.193	.350	.110	20.0	44.0	16,722
OpenOneRec-1.7B-pro	Both	.139	.184	.230	.370	.127	30.8	41.0	17,255
OpenOneRec-8B-pro	SID	.168	.228	.268	.455	.160	20.5	37.3	845
OpenOneRec-8B-pro	Text	.159	.226	.265	.480	.151	2.3	12.3	16,722
OpenOneRec-8B-pro	Both	.166	.222	.255	.428	.159	13.0	6.0	17,255

Table 1: Anchor results before any post-training (n=400, k=64, N=30). "Both" shows each item with both its SID and its caption. Input tokens are measured with the OpenOneRec tokenizer, one yardstick for every row; the APIs' own tokenizers count these prompts higher still (Appendix D.2). Identity% and Loop% are the interface diagnostics defined in Section 3. Random and Caption do not implement the reason-then-rank interface, so interface fields do not apply.

quality, because a model can fail the output format while still carrying the signal. RERANKID is the testbed we build to these requirements.

Problem formulation. For a user, let $H_k = (h_1, \ldots, h_k)$ be the k most recent history items and $C = (c_0, \ldots, c_{N-1})$ a slate of N candidates, exactly one of which is the observed positive c^+. An item rendering ρ maps an item to tokens, and the model reads the rendered pair $X_\rho = \big(\rho(H_k),\, \rho(C)\big)$, with $\rho \in \{\rho_{\text{sid}},\, \rho_{\text{text}},\, \rho_{\text{both}}\}$. Given X_ρ, the model reasons about the user's interests in natural language and returns a ranking $\pi = (\pi_1, \ldots, \pi_{10})$ of ten distinct candidate indices $\pi_j \in \{0, \ldots, N-1\}$, most relevant first. Ranking quality is scored by the rank of c^+ in π. Because π refers to candidates only by index, the same user, slate, and label can be scored under any rendering.

The three renderings are item-matched views of the same example: SID-only (ρ_{sid}), dense-caption text (ρ_{text}), and combined (ρ_{both}), which shows each item with both its SID and its caption. The primary setting is *pure-SID*: inference uses ρ_{sid} alone, so item titles, captions, categories, and derived user profiles are absent, while the task instruction and the generated rationale remain natural language. A gap between two views on the same users is then an item-rendering effect, not a model-family or population effect.

Controlled construction. We instantiate the testbed on Kuaishou short-video interactions and the OpenOneRec SID vocabulary (Zhou et al., 2026). Its SID tokens (*itemic tokens* in its terminology) quantize text embeddings of item descriptions, and their embeddings are aligned with text and then co-pretrained on user behavior sequences. Its matched public checkpoints, OpenOneRec-1.7B-pro and OpenOneRec-8B-pro, allow a scale comparison under one identifier system, so every comparison changes one factor at a time. The frozen test set contains 400 exact-one-positive users, balanced between explicit-engagement positives (a like, follow, or forward) and long-view positives, so an end-to-end score never averages over examples with ambiguous credit. Each user contributes one 100-item priority pool: the positive, clean exposure-hard negatives, and uniform-random catalog fill, with competing positives excluded rather than treated as negatives, so an item the user engaged with never enters a slate as a negative. Slates take the first N pool items under a frozen deterministic shuffle, so they are nested and growing N isolates the effect of added candidates. Histories use the same nested recent subsequence for every k, so the history curve measures added evidence for fixed users rather than a population shift.

From the anchor setting, k=64 and N=30, we vary one axis at a time: history length $k \in \{0, \ldots, 256\}$, slate size $N \in \{10, \ldots, 100\}$, and the item rendering (Figure 1). Two further display orders are reserved for a display-order audit. Evaluation users are disjoint from every post-training pool, and each history temporally precedes its target. This blocks instance-level leakage from our own training while deliberately retaining representation-level exposure: learning collaborative structure in the SID token space is the capability under study. Appendix A gives the eligibility

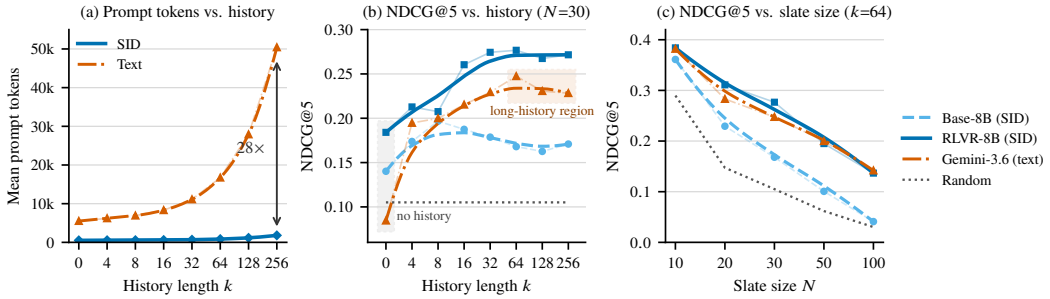

Figure 2: History and slate interventions. (a) Mean prompt tokens for the SID and text renderings as history grows, measured with the OpenOneRec tokenizer on the same users and slates. (b) NDCG@5 as history lengthens (N=30) and (c) as candidates are added (k=64), each varying one axis from the anchor setting, for the 8B Base, its RLVR policy, the Gemini-3.6 text reference, and seeded Random. Shaded boxes mark the no-history setting and the long-history region discussed in the text. Markers are measured values; curves are smoothed trends.

rules, label construction, negative recipe, and split manifests, and discusses pretraining exposure (Appendix A.4).

Scoring and diagnostics. The primary metric is end-to-end NDCG@5, with invalid outputs scored zero rather than dropped from a valid-only average, so interface failures count against a model; NDCG@10, Recall@5/10, and MRR are also reported. Because each user has exactly one observed positive, every reported metric is a function of that positive's returned rank. This is the standard offline protocol for implicit feedback (Kang & McAuley, 2018; Sun et al., 2019): the score asks whether a policy moves the observed item up, not whether the order among the remaining candidates is right, because no interaction log records that order. The audit in Section 5 separately judges how rationales argue the comparisons among candidates. Alongside accuracy we log each failure mode the task makes possible. Identity% counts rankings that return the candidate order exactly as shown, the literal ranking $[0, 1, \ldots, 9]$: under shuffled candidates this output passes the format check but carries no ranking decision (Wang & Lim, 2024; Liang et al., 2026; Liu & Li, 2026). Loop% counts responses whose reasoning never terminates and instead runs to the token cap. We also record final validity and use of the fixed extraction fallback. Main comparisons are paired over the same 400 users, summarized by a user-paired bootstrap interval for each method against its declared initializer; we call a difference resolved when its 95% interval excludes zero. One fixed prompt serves every system rather than per-model tuning (Yang et al., 2025; Wang et al., 2023). Both frozen choices are audited: rescoring under the reserved display orders and under natural rewordings of the prompt shifts the anchor score by no more than rerunning an unchanged condition does, and the text reference's top pick is statistically indistinguishable from uniform over the 30 display positions. Exact prompts, parsers, and budgets are in Appendix B, and both audits in Appendix D.3.

4 Diagnosing the Existing Models

We first evaluate the existing models before any post-training.

Setup. We evaluate the 1.7B and 8B OpenOneRec models on the frozen test set, reporting the anchor setting here and the interventions in Section 5.2. The two models are scored under all three renderings: SID-only is the primary condition, and the text and combined rows rerun the same users with readable item text. SID inference first permits free reasoning; a fixed forced-extraction stage is invoked only when no valid ranking is produced, and a response still invalid after that stage scores zero. Full protocol constants are in Appendix B.1.

Reference rows. The other rows of Table 1 are reference points. Random is a fixed seeded shuffle and sets the chance level. Caption ranks candidates by embedding similarity to the centroid of the history captions, a simple non-generative text baseline. GPT-5.6-Terra, Gemini-3.6-Flash,

and Claude-Sonnet-5 (OpenAI, 2026; Google DeepMind, 2026; Anthropic, 2026b) read the item-matched text view under the same task wording and answer format, and show what strong general models achieve when readable metadata is available. Their models and inputs both differ from OpenOneRec, so they serve as diagnostic references rather than representation-controlled competitors.

Ranking signal. At the anchor setting, the 8B model beats seeded Random from SID-only inputs with a resolved paired margin ($+.063$, CI $[+.027, +.099]$), while the 1.7B margin stays within noise ($+.009$, CI $[-.025, +.042]$); both remain below every text-view reference (Table 1). With no history at all (k=0, Figure 2b), the 8B margin over Random from the SID rendering of the candidates alone is $+.035$ (CI $[-.001, +.071]$), while the Gemini text reference is statistically indistinguishable from Random; post-training turns this margin into a resolved one (Section 5.2). Longer histories, in contrast, do not help. SID prompts stay a small fraction of the matched text prompts (Table 1, rightmost column), so extra history is nearly free for an SID model, yet as history grows the 8B model drifts with no sustained trend (Figure 2b).

Interface failures. The main obstacle is the interface, which is new to these models: no OpenOneRec training task presents a candidate slate and asks for an ordering. The models return the literal input order for a fifth to nearly half of users; in a typical case the 8B model closes an empty reasoning block after 46 tokens and emits $[0, 1, \ldots, 9]$, dropping a positive shown at position 29 (Appendix D.5, Case 1). Counting reasoning blocks that are nonempty and terminated, the 1.7B model completes a rationale for 1.0% of anchor users and the 8B model for 0.5%; the rest mostly leave the block empty or transcribe the prompt's history SIDs in order until the free-reasoning token cap (Case 2). The two-stage protocol still yields a valid list for every SID anchor example, but the extraction stage is invoked for 17.5% of anchor users at 1.7B and 37.5% at 8B, and scoring the free-reasoning stage alone would lower the SID Base scores from $.114$ to $.101$ and from $.168$ to $.113$ (Appendix B.1). Readable metadata does not remove the obstacle. Given captions instead of SIDs, the same models gain no consistent ranking accuracy (Table 1): the 8B scores stay within a few points across all three views, and the 1.7B model mostly trades identity rankings for reasoning loops. Captions change the failure mode, not the ranking accuracy.

5 Eliciting SID-Only Reasoning Reranking

We then ask which post-training signals elicit reasoning reranking, and how the elicited policies respond to the history and slate interventions.

5.1 Post-Training Setup

We compare four signals as experimental arms, each measured against its own initializer: SFT and OPD start from Base, while RFT and RLVR continue from the SFT checkpoint. They differ in where responses originate and how feedback enters: SFT imitates a text-view hindsight teacher, whereas OPD, RFT, and RLVR train on responses sampled by the SID student (Figure 1, right). No signal grades the rationale itself: the SFT loss covers the whole teacher trace, and the verifiers read only the final index list. We keep the rationale because it is what reasoning rerankers show a reader, and we evaluate its content directly (Section 5.2). Training users are disjoint from the evaluation users. Within each method, both scales share the data distribution, prompt, output contract, and per-update optimization settings; all arms use LoRA and leave SID embeddings unchanged. The paired intervals in Table 2 resample the evaluation users for each trained checkpoint. Exact recipes, training objectives, and compute are in Appendices C and E.

SFT. SFT asks whether caption-grounded hindsight demonstrations can teach an SID-only student a usable ranking policy. Gemini-3.6-Flash receives the caption rendering and the known positive, then writes a concise rationale and a ranked list; the student imitates that response from the item-matched SID rendering with next-token likelihood (Wang et al., 2024), over $11{,}031$ training demonstrations from users disjoint from evaluation. Quality gates enforce the output contract, positive-at-rank-one correctness, and candidate grounding, and answer-revealing phrasing is prohibited and audited. The teacher rationale is thus a hindsight training trace, and its ordering of the negatives is teacher-supplied.

Method	Init	Supervision	NDCG@5	ΔBase	ΔInit	95% CI vs. Init	Identity%	Loop%
OpenOneRec-1.7B-pro								
Base	–	–	.114	–	–	–	42.3	16.5
SFT	Base	Teacher traces	.225	+.111	+.111	$[+.071, +.151]$	0.0	0.0
OPD	Base	Teacher distributions	.195	+.082	+.082	$[+.044, +.120]$	0.0	0.0
RFT	SFT	Filtered rollouts	.223	+.109	-.002	$[-.028, +.023]$	0.0	0.0
RLVR	SFT	NDCG@5 reward	.231	+.117	+.006	$[-.015, +.026]$	0.0	0.0
RLVR	SFT	Hit@1 reward	.232	+.119	+.007	$[-.014, +.028]$	0.0	0.0
OpenOneRec-8B-pro								
Base	–	–	.168	–	–	–	20.5	37.3
SFT	Base	Teacher traces	.237	+.069	+.069	$[+.028, +.109]$	0.0	0.0
OPD	Base	Teacher distributions	.210	+.042	+.042	$[+.001, +.083]$	0.0	0.0
RFT	SFT	Filtered rollouts	.211	+.043	-.026	$[-.047, -.005]$	0.0	0.0
RLVR	SFT	NDCG@5 reward	.277	+.109	+.040	$[+.006, +.076]$	0.0	0.0
RLVR	SFT	Hit@1 reward	.232	+.064	-.004	$[-.025, +.016]$	0.0	0.0

Table 2: SID-only post-training at the anchor setting. Each block uses its own Base, and each supervision form is defined in Section 5.1. Confidence intervals are paired bootstrap intervals for NDCG@5 relative to the declared initializer. Repeated RLVR rows compare the graded and binary reward signals. Identity% and Loop% are the interface diagnostics of Table 1; NDCG@10, Recall@5/10, and MRR for every arm are in Table D.1. Arms differ in supervision form and budget by design, so each ΔInit is read against the arm's own initializer.

RFT. RFT asks whether successful responses the SFT student already samples can be amplified. For each SID-only prompt it draws eight responses, retains at most the first valid nonidentity trace that places the positive first, and continues likelihood training from the SFT checkpoint; this is offline verifier-filtered self-training in the style of STaR (Zelikman et al., 2022).

OPD. OPD asks whether dense guidance from a task-adapted policy can transfer reranking to the Base model on states the student itself visits (Agarwal et al., 2024). A fresh Base LoRA samples a response; the frozen SFT teacher supplies a next-token distribution on each student prefix; the update minimizes forward KL on the renormalized teacher top-32 support. Neither policy sees captions, gold positives, or any reward.

RLVR. RLVR asks whether the observed positive and a strict output contract supply enough direct policy-learning signal. Starting from SFT, GRPO (Shao et al., 2024) samples SID-only responses and assigns $R(y) = 1/\log_2(1 + r)$ when y is valid and returns the positive at rank $r \leq 5$, with ranks counted from one, and zero otherwise; a valid response must close its reasoning block and emit exactly ten distinct in-range indices. The optimized signal is exactly the primary NDCG@5 metric, with no auxiliary shaping, and training runs 100 optimizer steps at each scale. A controlled ablation replaces this reward with binary top-one correctness under the otherwise identical recipe.

5.2 Results

Main results. Table 2 separates the two scales and every initializer. SFT already teaches the interface: at both scales it eliminates every identity permutation and reasoning loop and adds a large, resolved ranking gain over Base. The extraction fallback nearly disappears with it; across the SFT, RFT, and RLVR arms it is invoked for at most 1.0% of anchor users and adds at most $+.004$ NDCG@5. The SFT anchor estimates come close to the Gemini text reference, and neither the remaining gap to Gemini nor the gap between the two SFT scales is resolved.

RLVR is the only arm that improves on SFT, and only at 8B, where the graded ranking reward adds a resolved paired gain and yields the strongest SID-only anchor estimate, .277, on par with the Gemini and GPT-5.6-Terra text references that read prompts about $20\times$ longer (paired differences $+.029$ and $+.021$, Appendix D.1). In hit-rate terms, it places the positive among its returned ten for 64.0% of users, against 57.0% for Gemini and 60.3% for GPT-5.6-Terra (Tables 1 and D.1). At 1.7B the same recipe lands within noise of SFT. The reward ablation follows the same pattern: the direct paired contrast puts the graded reward $+.045$ above binary top-one correctness at 8B (CI

$[+.013, +.077]$), while at 1.7B the two rewards are indistinguishable. Graded credit for near misses pays off only at the larger scale.

The other two arms stay at or below their supervised references. RFT amplifies verifier-accepted traces the SFT student already samples; it matches SFT at 1.7B and loses ground at 8B under the matched recipe. OPD receives no captions, no labels, and no reward, only teacher distributions on student-sampled SID trajectories, yet it still removes the interface failures and recovers most of the teacher's gain without exceeding it (74% at 1.7B and 61% at 8B). Every trained arm at both scales also ranks above the caption-centroid ranker (.184), which neither Base model reaches. Complete paired contrasts and validity diagnostics are in Appendix D.1.

History and slate size. At the anchor setting, the SID prompt averages 845 tokens against 16,722 for the item-matched text rendering; by k=256 the text prompt exceeds 50k tokens while the SID prompt stays under 2k, a 28× gap (Figure 2a). These counts use the OpenOneRec tokenizer for every rendering; the providers' own tokenizers count the anchor text prompt at 17.8k to 31.2k tokens, so the ratios are conservative. Over this range 8B RLVR climbs with history, with most of the climb in place by k=16, below the shortest history seen in training (32 items), and holds its plateau through k=256, a resolved paired gain over its own no-history setting (Figure 2b). The text curve comes from Gemini because the text prompt nears the 32k-token context of the OpenOneRec checkpoints at k=128 and exceeds it at k=256 (Zhou et al., 2026). The Gemini reference gains nothing past the anchor setting: it peaks there and drifts lower across the shaded region ($-.019$ by k=256, CI $[-.050, +.012]$) while its prompt passes 50k tokens. The Base model drifts with no sustained trend (Section 4), so the response to history comes from post-training. Histories are nested, so each larger k prepends the same user's earlier real interactions, shown as SIDs to one system and as captions to the other: the SID policy keeps its gains from this added evidence at lengths where the text reader stops improving. One user shows the pattern (Appendix D.5, Case 5). The history returns again and again to chat-record screenshots and emotional-confession videos, and the positive is another such video. From the candidates alone, 8B RLVR leaves it out of the returned ten; at k=64 and k=256 it ranks it second. Gemini ranks it first at k=64 but seventh at k=256, where its top pick switches to an outfit-change video of a celebrity who dominates one stretch of the added history. Added candidates lower every score, Random's included, since one positive now competes with more negatives (Figure 2c). The elicited 8B policy declines in parallel with the Gemini reference and ends level with it at N=100, while its Base collapses toward Random.

The no-history margin. With no history at all, the elicited policy beats Random by $+.079$ (CI $[+.039, +.119]$; Figure 2b, shaded setting). A k=0 prompt contains no history and no user identifier, so the margin can only come from the candidate SIDs, whose embeddings are learned in part from interaction sequences. On the same slates rendered as captions, the Gemini reference is statistically indistinguishable from Random. The SID policy therefore draws on item-level preference signal carried by these embeddings, which a strong general reader does not recover from the item text.

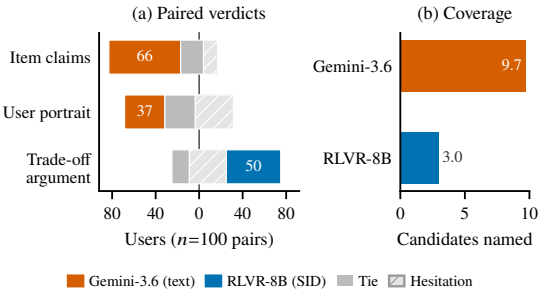

Figure 3: Paired rationale audit at the anchor setting (100 users; each pair judged in both presentation orders, and only order-consistent verdicts are decided). (a) Decided verdicts for three of the five judged aspects; hatched segments are hesitations, where the two presentation orders split between a win and a tie. Judged coverage and the overall verdict, which tracks it, go to Gemini (Table D.5). (b) Mean number of the ten ranked candidates that the rationale names, counted by the frozen parser.

Rationale quality. Since the elicited 8B policy ranks on par with the text references, we next compare its rationales with theirs. For 100 anchor users we take the rationales that 8B RLVR and the Gemini-3.6 reference actually generated during their frozen evaluations, verbatim, and have Claude-Opus-5 (Anthropic, 2026a) compare each pair aspect by aspect against the caption-rendered history and slate. Each pair is judged in both presentation orders, and a verdict counts only when both orders

agree. The judge grades item-level claims, the user portrait, the trade-off argument, coverage, and an overall verdict; coverage is also counted directly from the text. Appendix D.4 gives the protocol, counts, and judge validity checks.

The verdicts split by aspect (Figure 3). The SID rationales argue better: RLVR wins the trade-off aspect $50{:}0$, and it discusses its candidates in the order it ranks them (per-user Kendall $\tau{=}1.0$; $.983$ for Gemini, which discusses more of them). They describe worse: they name about three of the ten ranked candidates where Gemini names nearly all ten, item-level claims lose $0{:}66$, and the user portrait loses $0{:}37$. These errors follow from the input: a policy that has never read a caption can state the user's interest correctly yet attach it to an invented item. In one pair (Appendix D.5, Case 4), RLVR ranks the positive third and Gemini eighth, yet RLVR justifies its top pick as a recap of a novel whose title appears nowhere in the slate or the history. The candidate is a recap of a different novel, and the history does return repeatedly to novel recaps and emotional stories; Gemini describes the same candidate correctly in one line. The judge is stable: across the 500 paired aspect judgments, five per user pair, the decided direction reverses once, and on coverage it agrees with the programmatic count on every user where that count has a winner. Under SID-only input, ranking accuracy and rationale reliability therefore come apart: post-training taught the 8B policy to order the slate but not to describe what is in it.

6 Conclusion

We presented a controlled study of reason-then-rerank with SIDs as the sole item representation, on ReRankID, a frozen testbed built for this setting. The existing SID-pretrained models fail at the interface: they often return the input order unchanged, gain nothing from longer histories, and do not improve when given readable captions. Post-training elicits the behavior. SFT teaches the reason-then-rank interface at both scales, and at 8B, RLVR with a graded ranking reward improves on SFT, matches text-reading LLMs from prompts about $20\times$ shorter, ranks above random without any history, and keeps its gains at long histories where the text reference stops improving. The rationale audit locates what the tokens do not supply: the policy orders the slate well and argues its trade-offs sharply, but its claims about specific items are often wrong.

Read together, the three kinds of evidence locate what the SID tokens give a reasoning policy. They carry item-level preference signal that a strong text reader does not recover from captions: from the candidates alone, the SID policy beats Random while the caption reader does not. They make long histories usable: each added history item costs a few SID tokens instead of a caption, and the SID policy keeps its gains at lengths where the text reader stops improving. They do not carry enough detail to name items: the policy states interests the history supports but attaches them to items it cannot identify. A pure-SID reasoner is therefore suited to ordering a slate and poorly suited to describing it, which suggests a split design: rank from SIDs, which keeps long histories cheap and draws on behavioral signal, then write the explanation from retrieved metadata for the few items the ranking places on top. The testbed's text rendering also stands alone as a public evaluation of how well general LLMs model real user behavior.

AI Use Statement

In this work, we used generative AI models as components of the experimental method, as described in the paper: Gemini-3.6-Flash wrote the hindsight rationales and rankings that form the SFT training data (Section 5.1, Appendix C.1), Claude-Opus-5 served as the rationale judge (Section 5.2, Appendix D.4), and GPT-5.6-Terra, Gemini-3.6-Flash, and Claude-Sonnet-5 were evaluated as text-view reference systems (Section 4). We also used generative AI coding assistants to help implement the experiment code, including the benchmark builders, scoring and statistics scripts, post-training pipelines, and plotting code. Generative AI tools also helped draft and revise prose, restructure LaTeX source, and check formatting compliance. The authors reviewed all AI-assisted code, verified the scientific claims, experimental choices, data handling, figures, citations, and final manuscript, and take responsibility for all content of this work, including anything produced with the aid of generative AI.

Ethics Statement

The study uses the OpenOneRec research release of Kuaishou interaction data under its access terms; we collect no new data, and users appear only as opaque numeric identifiers. Prompts expose item captions and interaction histories, never user identifiers; caption text goes to commercial LLM APIs only for the text references, the teacher annotations, and the rationale judge. These calls use the providers' no-training, zero-retention settings. Printed excerpts translate user-facing text and redact personal names. The testbed measures engagement prediction for diagnostic purposes; deployment of its rankings would require platform-side review of fairness and exposure effects.

Reproducibility Statement

Appendix A specifies the frozen testbed: eligibility rules, the construction of users, slates, and negatives, the nested history and slate interventions, and the split sizes. Appendix B gives the metrics, the two-stage inference protocol, all decoding settings, the bootstrap procedure, and the exact evaluation and extraction prompts. Appendix C gives, for each post-training arm, the data construction, objective, hyperparameters, checkpoint-selection rule, and development trajectories, including the exact SFT teacher prompt. Appendix D reports the complete anchor metrics and paired contrasts, and Appendix E describes the anonymized release of data builders, split manifests, per-user result files, parsers, reward code, and run commands, together with compute and the number of training runs per configuration. The anonymized code is available at https://anonymous.4open.science/r/rerankid-code-8D2B.

References

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=3zKtaqxLhW.

Anthropic. System card: Claude Opus 5. https://www.anthropic.com/claude-opus-5-system-card, 2026a.

Anthropic. System card: Claude Sonnet 5. https://www.anthropic.com/claude-sonnet-5-system-card, 2026b.

Ethan Bito, Yongli Ren, and Estrid He. One pass, any order: Position-invariant listwise reranking for LLM-based recommendation. In Alistair Moffat, Falk Scholer, Hannah Bast, Marc Najork, and Min Zhang (eds.), *Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2026, MelbourneVICAustralia, July 20-24, 2026*, pp. 3625–3629. ACM, 2026. doi: 10.1145/3805712.3809952. URL https://doi.org/10.1145/3805712.3809952.

Wen-Shuo Chao, Zhi Zheng, Hengshu Zhu, and Hao Liu. Make large language model a better ranker. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 918–929, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.51. URL https://aclanthology.org/2024.findings-emnlp.51/.

Xiangyi Chen, Zelun Wang, Xinyi Li, Yi-Ping Hsu, Jaewon Yang, and Jiajing Xu. LLMs need encoders for semantic IDs too, 2026. URL https://arxiv.org/abs/2606.00324.

Revanth Gangi Reddy, JaeHyeok Doo, Yifei Xu, Md Arafat Sultan, Deevya Swain, Avirup Sil, and Heng Ji. FIRST: Faster improved listwise reranking with single token decoding. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 8642–8652, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.491. URL https://aclanthology.org/2024.emnlp-main.491/.

Google DeepMind. Gemini 3.6 Flash model card. https://deepmind.google/models/model-cards/gemini-3-6-flash/, 2026.

Yingzhi He, Yan Sun, Junfei Tan, Yuxin Chen, Xiaoyu Kong, Chunxu Shen, Xiang Wang, An Zhang, and Tat-Seng Chua. Reasoning over semantic IDs enhances generative recommendation. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2, KDD 2026*, pp. 1638–1649. ACM, 2026. doi: 10.1145/3770855.3818122.

Yupeng Hou, Junjie Zhang, Zihan Lin, Hongyu Lu, Ruobing Xie, Julian J. McAuley, and Wayne Xin Zhao. Large language models are zero-shot rankers for recommender systems. In Nazli Goharian, Nicola Tonellotto, Yulan He, Aldo Lipani, Graham McDonald, Craig Macdonald, and Iadh Ounis (eds.), *Advances in Information Retrieval - 46th European Conference on Information Retrieval, ECIR 2024, Glasgow, UK, March 24-28, 2024, Proceedings, Part II*, volume 14609 of *Lecture Notes in Computer Science*, pp. 364–381. Springer, 2024. doi: 10.1007/978-3-031-56060-6_24. URL https://doi.org/10.1007/978-3-031-56060-6_24.

Yupeng Hou, Haven Kim, Clark Mingxuan Ju, Eduardo Escoto, Neil Shah, and Julian McAuley. Expressiveness limits of autoregressive semantic ID generation in generative recommendation, 2026. URL https://arxiv.org/abs/2605.06331.

Wenyue Hua, Shuyuan Xu, Yingqiang Ge, and Yongfeng Zhang. How to index item IDs for recommendation foundation models. In Qingyao Ai, Yiqin Liu, Alistair Moffat, Xuanjing Huang, Tetsuya Sakai, and Justin Zobel (eds.), *Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region, SIGIR-AP 2023, Beijing, China, November 26-28, 2023*, pp. 195–204. ACM, 2023. doi: 10.1145/3624918.3625339. URL https://doi.org/10.1145/3624918.3625339.

Pengyue Jia, Xiaobei Wang, Yingyi Zhang, Shuchang Liu, Yupeng Hou, Hailan Yang, Xu Gao, Xiaopeng Li, Yejing Wang, Julian McAuley, Xiang Li, Lantao Hu, Yongqi Liu, Kaiqiao Zhan, Han Li, Kun Gai, and Xiangyu Zhao. From local indices to global identifiers: Generative reranking for recommender systems via global action space, 2026. URL https://arxiv.org/abs/2604.25291.

Wang-Cheng Kang and Julian J. McAuley. Self-attentive sequential recommendation. In *IEEE International Conference on Data Mining, ICDM 2018, Singapore, November 17-20, 2018*, pp. 197–206. IEEE Computer Society, 2018. doi: 10.1109/ICDM.2018.00035. URL https://doi.org/10.1109/ICDM.2018.00035.

Yufei Li, Zaiwei Zhang, Mingfu Liang, Kavosh Asadi, Jay Xu, Jimmy Kim, Chongyang Bai, Jieyi Zhang, Hongye Xie, Prachi Agrawal, Dian Yu, Tianyi Chen, Jean-Pascal Billaud, Garret Buell, Yongkang Zhu, Sachin Patil, Brooke Bian, Zhou Fang, Kevin Huang, Shiva Sudanagunta, Yuzhen Huang, Emma Lu, Chris O'Brien, Yang Song, Lihong Li, Jacob Tao, Zhicheng Zhu, Chao Li, Gaoxiang Liu, Neil Wu, Zhongyin Hu, Li Han, Loki Chen, Ming Lei, Greg Rehm, Siyuan Song, Tianwei Zhang, Li Li, Ketan Singh, Yavuz Yetim, Ilyas Atishev, Satendra Gera, Ashkan Sadeghi, Rachel Yan, Nikko Mizutani, Shuaiwen Wang, Song Yang, Zhijing Li, Jiang Liu, Mengying Sun, Fei Tian, Xiaohan Wei, Chonglin Sun, Shuo Gu, Parish Aggarwal, Senthil Manickavelu, Kaushik Rangadurai, Zhi Hua, Frank Shyu, Ruchit Sharma, Liyuan Li, Shike Mei, Wenlin Chen, Santanu Kolay, Ben Schulte, Deepak Chandra, Yang Song, Sandeep Pandey, Xi Liu, Hamed Firooz, and Luke Simon. GR2 technical report, 2026. URL https://arxiv.org/abs/2606.31984.

Mingfu Liang, Yufei Li, Jay Xu, Kavosh Asadi, Xi Liu, Shuo Gu, Kaushik Rangadurai, Frank Shyu, Shuaiwen Wang, Song Yang, Zhijing Li, Jiang Liu, Mengying Sun, Fei Tian, Xiaohan Wei, Chonglin Sun, Jacob Tao, Shike Mei, Wenlin Chen, Santanu Kolay, Sandeep Pandey, Hamed Firooz, and Luke Simon. GR2: Generative reasoning re-ranker, 2026. URL https://arxiv.org/abs/2602.07774.

Jiacheng Lin, Tian Wang, and Kun Qian. Rec-R1: Bridging generative large language models and user-centric recommendation systems via reinforcement learning. *Trans. Mach. Learn. Res.*, 2025, 2025. URL https://openreview.net/forum?id=YBRU9MV2vE.

Danyang Liu and Kan Li. Stop overthinking: Unlocking efficient listwise reranking with minimal reasoning, 2026. URL https://arxiv.org/abs/2605.14450.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Trans. Assoc. Comput. Linguistics*, 12:157–173, 2024. doi: 10.1162/TACL_A_00638. URL https://doi.org/10.1162/tacl_a_00638.

Zhanyu Liu, Shiyao Wang, Xingmei Wang, Rongzhou Zhang, Jiaxin Deng, Honghui Bao, Jinghao Zhang, Wuchao Li, Pengfei Zheng, Xiangyu Wu, Yifei Hu, Qigen Hu, Xinchen Luo, Lejian Ren, Zixing Zhang, Qianqian Wang, Kuo Cai, Yunfan Wu, Hongtao Cheng, Zexuan Cheng, Lu Ren, Huanjie Wang, Yi Su, Ruiming Tang, Kun Gai, and Guorui Zhou. OneRec-Think: In-text reasoning for generative recommendation. 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), ACL 2026, San Diego, California, United States, July 2-7, 2026*, pp. 2664–2681. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.123. URL https://aclanthology.org/2026.acl-long.123/.

OpenAI. GPT-5.6 system card. https://deploymentsafety.openai.com/gpt-5-6/gpt-5-6.pdf, 2026.

Ronak Pradeep, Sahel Sharifymoghaddam, and Jimmy Lin. RankZephyr: Effective and robust zero-shot listwise reranking is a breeze!, 2023. URL https://arxiv.org/abs/2312.02724.

Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan Hulikal Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Q. Tran, Jonah Samost, Maciej Kula, Ed H. Chi, and Mahesh Sathiamoorthy. Recommender systems with generative retrieval. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/20dcab0f14046a5c6b02b61da9f13229-Abstract-Conference.html.

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.

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 Wenwu Zhu, Dacheng Tao, Xueqi Cheng, Peng Cui, Elke A. Rundensteiner, David Carmel, Qi He, and Jeffrey Xu Yu (eds.), *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*, pp. 1441–1450. ACM, 2019. doi: 10.1145/3357384.3357895. URL https://doi.org/10.1145/3357384.3357895.

Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. Is ChatGPT good at search? investigating large language models as re-ranking agents. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 14918–14937, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.923. URL https://aclanthology.org/2023.emnlp-main.923/.

OneRec Team, Biao Yang, Boyang Ding, Chenglong Chu, Dunju Zang, Fei Pan, Han Li, Hao Jiang, Honghui Bao, Huanjie Wang, Jian Liang, Jiangxia Cao, Jiao Ou, Jiaxin Deng, Jinghao Zhang, Kun Gai, Lu Ren, Peiru Du, Pengfei Zheng, Rongzhou Zhang, Ruiming Tang, Shiyao Wang, Siyang Mao, Siyuan Lou, Teng Shi, Wei Yuan, Wenlong Xu, Xingchen Liu, Xingmei Wang, Xinqi Jin, Yan Sun, Yan Wang, Yifei Hu, Yingzhi He, Yufei Ye, Yuhao Wang, Yunhao Zhou, Yuqin Dai, Zhao Liu, Zhipeng Wei, Zhixin Ling, Ziming Li, Zixing Zhang, Ziyuan Liu, An Zhang, Changxin Lao, Chaoyi Ma, Chengru Song, Defu Lian, Fan Yang, Guowang Zhang, Hao Peng, Jiayao Shen, Jie Chen, Jun Xu, Junmin Chen, Kun Zhang, Kuo Cai, Mingxing Wen, Minmao Wang, Minxuan Lv, Qi Zhang, Qiang Luo, Sheng Yu, Shijie Li, Shijie Yi, Shuang Yang, Shugui Liu, Shuni Chen, Tinghai Zhang, Tingting Gao, Xiang Wang, Xiangyu Wu, Xiangyu Zhao, Xiao Lv, Xiaoyou Zhou, Xuming Wang, Yong Du, Zejian Zhang, Zhaojie Liu, Zhiyang Zhang, Zhuang Zhuang, Ziqi Wang, and Ziyi Zhao. OneReason technical report, 2026. URL https://arxiv.org/abs/2606.06260.

Lei Wang and Ee-Peng Lim. The whole is better than the sum: Using aggregated demonstrations in in-context learning for sequential recommendation. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 876–895, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-naacl.56. URL https://aclanthology.org/2024.findings-naacl.56/.

Xinfeng Wang, Jin Cui, Yoshimi Suzuki, and Fumiyo Fukumoto. RDRec: Rationale distillation for LLM-based recommendation. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 65–74, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-short.6. URL https://aclanthology.org/2024.acl-short.6/.

Yu Wang, Zhiwei Liu, Jianguo Zhang, Weiran Yao, Shelby Heinecke, and Philip S. Yu. DRDT: Dynamic reflection with divergent thinking for LLM-based sequential recommendation, 2023. URL https://arxiv.org/abs/2312.11336.

Orion Weller, Kathryn Ricci, Eugene Yang, Andrew Yates, Dawn Lawrie, and Benjamin Van Durme. Rank1: Test-time compute for reranking in information retrieval, 2025. URL https://arxiv.org/abs/2502.18418.

Wujiang Xu, Qitian Wu, Zujie Liang, Jiaojiao Han, Xuying Ning, Yunxiao Shi, Wenfang Lin, and Yongfeng Zhang. SLMRec: Distilling large language models into small for sequential recommendation. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=G4wARwjF8M.

Eugene Yang, Andrew Yates, Kathryn Ricci, Orion Weller, Vivek Chari, Benjamin Van Durme, and Dawn Lawrie. Rank-K: Test-time reasoning for listwise reranking, 2025. URL https://arxiv.org/abs/2505.14432.

Runyang You, Yongqi Li, Xinyu Lin, Xin Zhang, Wenjie Wang, Wenjie Li, and Liqiang Nie. R^2ec: Towards large recommender models with reasoning. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025*, pp. 69380–69409, 2025. doi: 10.52202/085713-2088.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. STaR: Bootstrapping reasoning with reasoning. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/639a9a172c044fbb64175b5fad42e9a5-Abstract-Conference.html.

Le Zhang, Bo Wang, Xipeng Qiu, Siva Reddy, and Aishwarya Agrawal. REARANK: Reasoning re-ranking agent via reinforcement learning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 2458–2471, Suzhou, China, November 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.emnlp-main.125. URL https://aclanthology.org/2025.emnlp-main.125/.

Zhuoxuan Zhang, Kangqi Ni, Yuhang Chen, Mingfu Liang, Xiaohan Wei, Yunchen Pu, Fei Tian, Chonglin Sun, Frank Shyu, Adam Yang Song, Sandeep Pandey, Luke Simon, Tianlong Chen, and Xi Liu. Diffusion-GR2: Diffusion generative reasoning re-ranker, 2026. URL https://arxiv.org/abs/2607.01170.

Kehan Zheng, Deyao Hong, Qian Li, Jun Zhang, Huan Yu, Jie Jiang, and Hongning Wang. Reasoning to rank: An end-to-end solution for exploiting large language models for recommendation, 2026a. URL https://arxiv.org/abs/2602.12530.

Zaiyi Zheng, Guanghui Min, Yaochen Zhu, Liang Wu, Liangjie Hong, Chen Chen, and Jundong Li. SAPO: Step-aligned policy optimization for reasoning-based generative recommendation, 2026b. URL https://arxiv.org/abs/2605.17648.

Guorui Zhou, Honghui Bao, Jiaming Huang, Jiaxin Deng, Jinghao Zhang, Junda She, Kuo Cai, Lejian Ren, Lu Ren, Qiang Luo, Qianqian Wang, Qigen Hu, Rongzhou Zhang, Ruiming Tang, Shiyao Wang, Wuchao Li, Xiangyu Wu, Xinchen Luo, Xingmei Wang, Yifei Hu, Yunfan Wu, Zhanyu Liu, Zhiyang Zhang, Zixing Zhang, Bo Chen, Bin Wen, Chaoyi Ma, Chengru Song, Chenglong Chu, Defu Lian, Fan Yang, Feng Jiang, Hongtao Cheng, Huanjie Wang, Kun Gai, Pengfei Zheng, Qiang Wang, Rui Huang, Siyang Mao, Tingting Gao, Wei Yuan, Yan Wang, Yang Zhou, Yi Su, Zexuan Cheng, Zhixin Ling, and Ziming Li. OpenOneRec technical report, 2026. URL https://arxiv.org/abs/2512.24762.

Appendix Overview

Appendices A and B specify the benchmark and the evaluation protocol, including the exact prompts. Appendix C specifies the four post-training recipes. Appendix D collects the additional results: the remaining anchor metrics, paired contrasts, diagnostic breakdowns, robustness audits, and qualitative cases. Appendix E describes the released artifacts and compute.

A Benchmark Construction

A.1 Task and Renderings

Let $H = (h_1, \ldots, h_T)$ be a user's chronologically ordered history and $C = (c_0, \ldots, c_{N-1})$ a candidate slate. Given the last k eligible history items and all N candidates, a model returns a rationale and an ordered JSON array of ten distinct candidate indices. The primary condition renders every item only as its three-codebook Semantic ID (SID). The item-matched text and combined views change only the item rendering: text replaces each SID with a dense caption, and the combined view shows each item with both its SID and its caption. Users, items, labels, display order, and the index-valued action space remain fixed. The phrase "SID-only" refers to the student's inference-time item view. It does not imply that post-training supervision contains no text.

A.2 Users and Slates

The source contains 162,074 users. An evaluation user must have a renderable positive and at least 256 history items that have both SID and caption views. The development and test splits each contain 400 users, with 200 exact-one grade-3 users and 200 exact-one grade-2 users. A grade-3 positive is a like, follow, or forward. A grade-2 positive is a long-view event for a user with no competing grade-3 event. No user crosses the SFT, shared post-training, benchmark-development, or benchmark-test partitions, and every history temporally precedes its target.

For each evaluation user, a 100-item priority pool contains the positive, clean exposure-hard negatives, and uniform-random catalog fill. Additional long-view items for grade-3 users are excluded rather than labeled negative. Candidate sets are nested at $N \in \{10, 20, 30, 50, 100\}$. Histories are nested at $k \in \{0, 4, 8, 16, 32, 64, 128, 256\}$ and retain the same recent unified items under every rendering. Display order is a deterministic function of user, slate size, and shuffle replicate. The anchor is $k = 64$, $N = 30$, and display seed zero. Table A.1 summarizes the contract.

A.3 Label Construction

Each example contains one observed positive, so NDCG@5, NDCG@10, Recall, and MRR are all functions of that positive's returned rank, the standard offline protocol for implicit feedback. Two constructions keep the labels clean. Competing positives are excluded from the candidate pool rather

Setting	Value
Platform / SID vocabulary	Kuaishou short-video interactions / OpenOneRec
Development / test	400 / 400 user-disjoint examples
Positive strata per split	200 grade-2 and 200 grade-3 users
Labels per example	Exactly one behaviorally observed positive
Minimum renderable history	256 items with both SID and caption views
History intervention	$k \in \{0, 4, 8, 16, 32, 64, 128, 256\}$, nested recent subsequences
Slate intervention	$N \in \{10, 20, 30, 50, 100\}$, nested from one 100-item pool
Negative construction	Clean exposure-hard negatives followed by uniform-random catalog fill
Output	Ten distinct in-range candidate indices, ordered by predicted relevance
Anchor	$k = 64$, $N = 30$, display seed zero
Test diagnostics	Mean 6.6 exposure-hard negatives; median unified history length 477
Coverage	100% SID and caption coverage in every evaluation slate

Table A.1: Frozen benchmark construction. One intervention axis varies at a time from the anchor setting while users, labels, and the other axis remain fixed.

than treated as negatives (Section A.2), so an item the user engaged with never enters a slate as a negative. The highest-priority negatives are exposure-hard: the platform showed the item and the user did not engage, which is behavioral evidence against it. Every system ranks the same frozen slates under the same labels, and the paper's claims rest on paired per-user comparisons. Comparisons among the candidates are also examined as argued text by the rationale audit of Section D.4.

A.4 Pretraining Exposure

The evaluation users come from the OpenOneRec public release, part of the platform interaction data on which the released checkpoints are pretrained (162,074 users in the dump we build from). What the backbone knows about an SID token is learned in this pretraining, which includes interaction sequences involving its item; this representation-level knowledge is the object of study.

Three structural facts separate this from example-level memorization, the concern that a checkpoint recalls that a specific history continues with a specific target. First, no prompt in the benchmark contains a user identifier, and the $k{=}0$ setting contains no history either, so at $k{=}0$ a memorized user–item association has no key to look up; the no-history margin is measured exactly where this channel is closed, and only item-level signal remains expressible. Second, post-training cannot install test-set memories, because its users are disjoint from the evaluation users. Third, the elicited policy's anchor accuracy is on par with the strongest text reference, a model with no exposure to this platform reading the same evidence as captions, so the accuracy level itself does not require a memorization explanation.

The training tasks of the release also differ in kind from the task studied here. They include catalog-level next-item generation, pointwise label prediction over a single candidate, and recommendation explanation, but no task presents a candidate slate and asks for a reasoned ordering. The reranking interface is therefore new to the checkpoints.

A.5 Scope of the Reference Set

The reference set calibrates the quantities the study varies. Random calibrates the score scale of the frozen slates. Caption calibrates what a non-generative text signal achieves on the same users. The three API references calibrate strong general text readers under the same task wording and answer format. Every reference is scored on the same frozen slates as the systems under study, so each headline comparison stays paired.

B Evaluation Protocol

B.1 Metrics and Inference

NDCG@5 is primary. We additionally report NDCG@10, Recall@5, Recall@10, and MRR. A valid final answer contains exactly ten distinct integer indices in range. An example still invalid after the complete inference protocol receives zero. Interface diagnostics include final validity, the literal identity ranking $[0, 1, \ldots, 9]$, the stage-1 reasoning-loop rate, and use of the extraction fallback. Valid-only averages are never substituted for the end-to-end scores.

SID inference uses a fixed two-stage protocol. Stage 1 permits free reasoning with greedy decoding. If it does not produce a valid list, stage 2 receives up to the first 1,200 characters of that reasoning and uses structured decoding to extract the list. Both stages and every failure are retained. The reported final validity is measured after this fixed fallback, whereas the loop statistic describes stage 1. Text API references use the same task and index contract with structured output.

Table B.1 accounts for the fallback at the anchor setting. Extraction% is the share of users for whom stage 1 produced no valid list, so stage-1 validity is its complement. The stage-1-only column rescores each system with every extraction-salvaged response set to zero, isolating what the fallback contributes to the end-to-end score. Before post-training the fallback carries real weight: it supplies $+.055$ of the 8B SID Base score and up to $+.065$ under the combined rendering at 1.7B. The dependence is heavier with no history: at $k{=}0$ the stage is invoked for 47.0% of users at 1.7B and 19.5% at 8B, and stage-1-only scoring would lower the $k{=}0$ SID Base scores from .119 to .053

System	View	Extraction%	Final valid%	NDCG@5	Stage-1 only
Base 1.7B	SID	17.5	100.0	.114	.101
Base 1.7B	Text	47.8	99.5	.115	.071
Base 1.7B	Both	43.5	99.8	.139	.074
Base 8B	SID	37.5	100.0	.168	.113
Base 8B	Text	13.0	99.8	.159	.140
Base 8B	Both	6.2	100.0	.166	.154
SFT 1.7B	SID	0.5	100.0	.225	.225
SFT 8B	SID	0.5	100.0	.237	.234
RFT 1.7B	SID	0.2	100.0	.223	.223
RFT 8B	SID	0.0	100.0	.211	.211
RLVR (NDCG@5) 1.7B	SID	0.2	100.0	.231	.231
RLVR (NDCG@5) 8B	SID	0.8	100.0	.277	.276
RLVR (Hit@1) 1.7B	SID	0.2	100.0	.232	.232
RLVR (Hit@1) 8B	SID	1.0	100.0	.232	.228

Table B.1: Forced-extraction accounting at the anchor setting. Stage-1 validity is the complement of Extraction%. Stage-1 only rescores the system with extraction-salvaged responses set to zero.

Setting	Value
SID models	OpenOneRec/OneRec-1.7B-pro and OpenOneRec/OneRec-8B-Pro, pinned released revisions
Text references	Gemini-3.6-Flash and Claude-Sonnet-5 through Vertex AI; GPT-5.6-Terra through OpenRouter; retained finish reason
Shared top-K	10 candidate indices
SID stage 1	Greedy decoding, 2,048-token response budget
SID stage 2	Structured array extraction, 256-token budget, 1,200-character reasoning cap
Text API decoding	Gemini: visible rationale, medium thinking, provider-default temperature, 8,192-token cap; Claude and GPT: 32,768-token cap
Primary metric	End-to-end NDCG@5, with outputs still invalid after extraction scored zero
Additional metrics	NDCG@10, Recall@5/10, MRR
Interface diagnostics	Final validity, stage-1 loop, extraction use, literal identity ranking
Uncertainty	10,000 user-paired bootstrap resamples; seed zero
Calibration rankers	Seeded Random (seed 7); Caption uses Qwen3-Embedding-0.6B centroid similarity

Table B.2: Evaluation and inference settings.

and from .140 to .119. After post-training every SFT, RFT, and RLVR arm invokes it for at most 1.0% of users and gains at most $+.004$ from it.

All headline comparisons use the same 400 users. The primary uncertainty summary is a percentile 95% confidence interval from 10,000 user-level paired bootstrap resamples of the method-minus-initializer NDCG@5 difference. Table B.2 lists the inference settings.

B.2 Shared Evaluation Prompt

The only system-prompt substitution across renderings is the representation noun: “Semantic ID”, “text description”, or “Semantic ID and its text description”. The faithful English translation comes first, and the exact UTF-8 production prompt follows it. Bracketed spans in the translation and brace-delimited placeholders in the original mark row-specific data fields, not extra instructions. Natural rewordings of this prompt are audited in Section D.3.

Shared evaluation prompt (English translation)

```
SYSTEM: You are a recommendation reranking engine.  Given the known
user information, which may be empty, and a set of candidate items
represented as [REPRESENTATION], rank the candidates from most to
least relevant according to the user's interests.

USER: [CHRONOLOGICAL USER HISTORY, OR AN EXPLICIT STATEMENT THAT NO
HISTORY IS AVAILABLE]

Candidates (N items, numbered 0 through N-1):
[0] [RENDERED CANDIDATE 0]
...
[N-1] [RENDERED CANDIDATE N-1]

Briefly analyze the user's interests and how well the candidates
match them.  Refer to items by candidate number, for example
``candidate 7''.  After the reasoning ends, select the ten candidates
that you consider most relevant and order them from most to least
relevant.  Output a JSON array containing exactly ten distinct
candidate indices in the range 0 through N-1, in the form [<most
relevant index>, <second most relevant index>, ...].
```

Shared evaluation prompt (exact UTF-8 original)

```
SYSTEM: 你是一个推荐重排序引擎。给定用户的已知信息（可能为空）和一组用语义ID表示
的候选物品，请根据用户兴趣把候选按相关性从高到低排序。

USER: {USER_CONTEXT}

候选物品（共 30 个，编号 0 到 29）:
{30 NUMBERED SID CANDIDATES}

请简明扼要地分析用户兴趣与各候选的匹配程度，分析时请用候选编号（如「候选7」）指代物
品，然后在思考结束后，从这 30 个候选中挑出你认为最相关的 10 个，按相关性从高到低
排列（最相关在最前）。输出一个 JSON 数组，恰好包含 10 个互不相同的候选编号（取值 0
到 29），形如 [<最相关的编号>, <次相关的编号>, ···]。
```

The schematic output format contains no literal demonstration indices, which prevents a weak model from copying an example ranking. When stage 1 fails, the extraction stage uses the system message "You are a ranking-result extractor: read a ranking analysis and output the final index ordering." Its user message restates N and the ten-index contract, supplies the capped stage-1 analysis, and requests only one JSON array with the most relevant index first. This extraction prompt is fixed across all SID policies.

Extraction-stage prompt (exact UTF-8 original)

```
SYSTEM: 你是一个排序结果抽取器：读取一段排序分析，输出最终的编号排列。

USER: 任务：从 30 个候选（编号 0 到 29）中选出最相关的 10 个，并从高到低排序。
以下是之前的分析过程：
{STAGE-1 REASONING}

请只输出一个 JSON 数组，恰好包含 10 个互不相同的编号（取值 0 到 29），最相关在前。
```

The reported Gemini-3.6 reference uses the same text-rendered task prompt with an explicit request to place its visible analysis inside `<think>...</think>` before the index array. We disable response-schema forcing so that this rationale is retained, use medium thinking with the provider-default temperature, and allow 8,192 output tokens. At the anchor setting, 399 of 400 outputs satisfy the ranking contract and all 400 contain a nonempty visible rationale.

Setting	Value
Source / quality-accepted examples	12,000 / 11,991 unique users
Source positive strata	Grade 2: 6,000; grade 3: 6,000
Accepted positive strata	Grade 2: 5,994; grade 3: 5,997
Train / development	11,031 / 960 accepted examples, with zero overlap
Quality rejections	9 total: grounding 4; length 4; correctness 1
Source-pool history counts	$k = 32$: 1,179; 64: 3,648; 128: 3,692; 256: 3,481
Source-pool slate counts	$N = 10$: 1,225; 20: 1,761; 30: 4,816; 50: 2,385; 100: 1,813
Teacher	Gemini-3.6-Flash, Vertex AI, medium thinking, provider-default temperature
Teacher response budget	8,192 tokens; one format-correction retry
Teacher view / student view	Item-matched caption / SID-only
Acceptance gates	Exact ten-index format, observed positive at rank one, explicit grounding to that candidate, nonempty rationale at most 600 characters
Answer-reveal handling	Prohibited by the teacher prompt and audited
Behavioral labels	One observed positive; negative-tail order supplied by the teacher

Table C.1: Final SFT annotation and data settings. No other SFT data recipe is part of the reported result.

C Post-Training Recipes

The four post-training arms share the SID-only task and evaluation benchmark. SFT starts from each Base model, whereas RFT and RLVR start from the corresponding selected SFT checkpoint. OPD starts from Base and queries a frozen, scale-matched SFT teacher on student-generated SID trajectories. Each arm is defined by its initializer, supervision, and budget, and is compared against its own initializer.

RFT, OPD, and RLVR draw on one shared 12,000-user post-training pool that is user-disjoint from SFT, benchmark development, and benchmark test. Their common 6,000-prompt training set is balanced by positive stratum and distributed as $k = 32$: 582, $k = 64$: 1,785, $k = 128$: 1,820, $k = 256$: 1,813 and $N = 10$: 580, $N = 20$: 905, $N = 30$: 2,457, $N = 50$: 1,167, $N = 100$: 891. We refer to this below as the shared post-training prompt set. All checkpoint selection uses benchmark-development NDCG@5 at the anchor setting; test data is never consulted before the final evaluation.

C.1 Supervised Fine-Tuning

The SFT source pool contains 12,000 unique users, with 6,000 examples from each positive stratum and one slate per user. We sample only long-history conditions, using $k \in \{32, 64, 128, 256\}$ with probabilities .10, .30, .30, .30. Candidate counts follow $N \in \{10, 20, 30, 50, 100\}$ with probabilities .10, .15, .40, .20, .15. The realized k and N counts in Table C.1 describe this 12,000-example source pool. Quality checks reject nine annotations, leaving 11,991 accepted examples. The final split contains 11,031 training and 960 development examples.

The caption teacher receives a privileged field containing the observed positive index. It writes a hindsight rationale and ranks that candidate first. The student then receives the item-matched SID-only prompt and imitates the teacher response with ordinary conditional next-token likelihood,

$$L_{\mathrm{SFT}}(\theta) = -\sum_{(x,\, y^*) \in \mathcal{D}_{\mathrm{SFT}}} \sum_{t=1}^{|y^*|} \log p_\theta(y_t^* \mid x,\, y_{<t}^*)\,, \tag{C.1}$$

where x is the SID-only prompt, y^* is the accepted teacher response, rationale followed by the index array, and θ are the LoRA parameters. The teacher rationale is therefore a hindsight training trace. The complete teacher instruction is translated below and followed by the exact UTF-8 production prompt. Table C.2 gives the optimization settings.

SFT hindsight-teacher prompt (English translation)

```
SYSTEM: You are an assistant that simulates the internal reasoning
process of a recommendation reranking model.  You will see user
context, numbered candidates rendered as text descriptions, and one
truly relevant candidate index supplied only for internal
construction.  Based on the user's history, write a concise analysis
that reads like an independent judgment and provide a top-10 index
list.  Never mention an internal reference, correct answer, reference
answer, being told, ground truth, or answer key.  Refer to items only
as ``candidate X''. Do not copy candidate descriptions or enumerate
every candidate at length.

USER: [TEXT-RENDERED USER CONTEXT]

Candidates (N items, numbered 0 through N-1):
[0] [CANDIDATE 0 DESCRIPTION]
...
[N-1] [CANDIDATE N-1 DESCRIPTION]

Relevant candidate index supplied for internal supervision:  [GOLD
INDEX]. The other candidates have no observed positive feedback, but
this does not mean that they are semantically unrelated.  Still
distinguish their degrees of relevance from the user's history.

Writing requirements:
1.  In three to five concise sentences, summarize interests supported
by the history and explain why the first-ranked candidate matches.
Keep the language proportional to the evidence.  Avoid exaggerated or
absolute claims such as ``extremely'', ``precisely'', ``highly'',
``strongly'', ``perfectly'', or ``inevitably''.  When evidence is
sparse, use restrained language such as ``may'' or ``to some
extent''.
2.  Consider both recent interests and interests in the longer
history.  Only when the history supports it, distinguish a currently
prominent interest, a stable interest that recurs over the longer
history, and an incidental topic.  Do not invent preferences merely
to cover both recent and long-term evidence, or describe a rare topic
as stable.  History items have no candidate numbers, so do not invent
labels such as ``history X''.
3.  Explicitly mention the first-ranked item as ``candidate X'' and
briefly compare one or two alternatives to explain why the first
choice fits better.  If one alternative is discussed, it must be
second in the final array.  If two are discussed, they must be second
and third.  The judgments in the analysis must agree with the final
ranking.  Do not enumerate every candidate or copy the full top-10
list into the analysis.
4.  Judge all candidates by semantic fit to the complete user
history.  Put the observed relevant candidate first.  Rank every
remaining candidate by its own fit to the history, not merely by
similarity to the first choice.  Before ranking, check for major
interests supported by multiple history events.  If a candidate
matches such an interest, place it appropriately even if its topic
differs from the first choice.  Do not allocate positions
mechanically across topics.
5.  The response must begin with <think>.  Put a nonempty analysis
between <think> and </think>, then start a new line and output one
JSON array.  The array must contain exactly ten distinct indices in
the range 0 through N-1.  The complete ranking may appear only once
and only after </think>.  An array without analysis, or before the
analysis, is invalid.

FORMAT-CORRECTION SUFFIX, USED ONCE IF REQUIRED: Independently redo
the task.  The response must begin with <think>, end a nonempty
analysis with </think>, and then output exactly one JSON array
containing ten distinct, in-range candidate indices.  Do not add
another tag, heading, or array.
```

SFT hindsight-teacher prompt (exact UTF-8 original)

SYSTEM: 你是一个模拟推荐重排序模型内部推理过程的助手。你会看到用户上下文、若干个带编号的候选，候选用文字描述呈现，以及仅供内部使用的一个真实相关候选编号。请根据用户历史写出读起来像独立判断的简明分析，并给出 top-10 编号列表。不要提及内部参考、正确答案、参考答案、被告知、ground truth 或 answer key。分析时只用「候选X」指代物品，不要复述候选描述，也不要逐项长篇罗列。

USER: {CAPTION-RENDERED USER CONTEXT}

候选物品（共 30 个，编号 0 到 29）：
{30 NUMBERED CAPTION CANDIDATES}

供内部构造监督使用的相关候选编号：[{GOLD INDEX}]。其余候选没有观察到正反馈，但这不代表它们语义上完全不相关；仍需根据用户历史区分其相关程度。

写作要求：
1. 用简洁的 3-5 句话概括有历史证据支持的用户兴趣，并说明首选候选的匹配依据，不要写得过长。措辞应与历史证据的强度相称，避免使用「极其」「精准」「高度」「强烈」「完美」「必然」等夸张或绝对化判断；证据较少时使用「可能」或「一定程度」等克制措辞。
2. 同时考虑近期兴趣和更长历史中的兴趣，但只在历史证据支持时区分近期突出兴趣、长历史中反复出现的稳定兴趣和偶然主题。不要为了覆盖近期与长期而强行编造偏好，也不要把少量出现的主题描述成稳定兴趣。历史记录本身没有候选编号，不要编造「历史X」之类的历史编号。
3. 用「候选X」明确提到首选候选，并简要比较一到两个备选候选，说明首选为什么更合适。若比较一个，它必须是最终数组中的第二名；若比较两个，它们必须恰好是最终数组中的第二名和第三名。分析中的判断必须与最终排序一致。不要逐个候选罗列，也不要在分析中列出或复述完整 top-10。
4. 最后按与用户完整历史的语义匹配程度对所有候选进行判断。把相关候选排在第一位；除首选外，其余候选也应分别按照它们与完整用户历史的匹配程度排序，不要仅按照它们与首选候选的相似程度排序。排序前检查历史中有多条记录支持的主要兴趣：若候选中有相应物品，应按证据纳入合适位置，不要因为它与首选主题不同而忽略；也不要为不同兴趣机械分配名额。

5. 回复必须以 `<think>` 开头。先输出 `<think>` 和 `</think>` 包围的非空分析，再另起一行输出 JSON 数组。数组必须恰好包含 10 个互不相同且范围在 0 到 29 的编号，且完整排序只能在 `</think>` 后出现一次。只输出数组或把数组放在分析前均无效。

FORMAT-CORRECTION SUFFIX, USED ONCE IF REQUIRED: 输出校正：请重新独立完成上述任务。回复必须严格以 `<think>` 开始，以 `</think>` 结束非空分析，然后只输出一次包含 10 个互不相同且范围合法的候选编号的 JSON 数组，不要添加其他标签、标题或数组。

Setting	1.7B Pro	8B Pro
Initialization	Base model	Base model
Adaptation	LoRA $r = 16$, $\alpha = 32$, all linear layers	Same
Frozen weights	Base parameters, including SID embeddings	Same
Template / cutoff	Qwen3 / 8,192 tokens	Qwen3 / 4,096 tokens
Precision	bf16	bf16
Effective batch	32	128
Learning rate / schedule	10^{-4} / cosine	5×10^{-5} / cosine
Warmup	.08	.08
Training epochs	3	3
Checkpoint interval	0.5 epoch	0.5 epoch
Selection criterion	Benchmark-dev NDCG@5 at $k = 64, N = 30$	Same
Selected epoch	3.0	1.01

Table C.2: Final SFT optimization settings. Both scales use the same accepted examples (11,031 training, 960 development). The 8B hyperparameters are selected by benchmark-development NDCG@5 before the test evaluation.

N	Source	1.7B retained	1.7B coverage	8B retained	8B coverage
10	580	360	62.07%	315	54.31%
20	905	373	41.22%	354	39.12%
30	2,457	826	33.62%	843	34.31%
50	1,167	267	22.88%	274	23.48%
100	891	114	12.79%	126	14.14%

Table C.3: RFT retention by candidate count from the shared 6,000-prompt source population. At most one response is retained per source prompt.

Setting	Value
Models	OpenOneRec 1.7B Pro and 8B Pro
Initialization	Corresponding dev-selected SFT checkpoint
Source prompts	Shared post-training prompt set (6,000 SID-only examples)
Samples per prompt	8
Sampling	Temperature 1.0, top-p .95, seed zero
Response budget	2,048 tokens
Retention gate	First completed, valid, nonidentity response with positive at rank one; at most one per prompt
Accepted traces	1.7B: 1,940; 8B: 1,912
Update	Continue LoRA $r = 16$, $\alpha = 32$, all linear layers
Cutoff / precision	8,192 tokens / bf16
Effective batch	32
Learning rate / schedule	5×10^{-5} / cosine, warmup .03
Training epochs	2
Checkpoint interval	0.5 epoch
Selection criterion	Highest benchmark-dev NDCG@5 at $k = 64$, $N = 30$
Selected checkpoint	1.7B: step 62; 8B: step 90

Table C.4: Frozen RFT settings. Both model scales use the same prompt keys, sampling hyperparameters, verifier, and update hyperparameters, while their sampled responses and accepted datasets remain separate.

C.2 Rejection-Sampling Fine-Tuning

RFT is an offline self-training baseline initialized from SFT and trained on the shared post-training prompt set. For each SID-only prompt, the frozen SFT policy samples eight responses. The policy never sees the positive label. An offline verifier scans responses in generation order and retains at most the first response that closes its reasoning, returns a valid nonidentity list, and places the observed positive at rank one. A response is rejected for an invalid or unclosed answer, generation-length truncation, an empty rationale, a positive below rank one, or identity collapse. The resulting 1.7B dataset contains 1,940 traces, corresponding to 32.33% prompt coverage. The matched 8B dataset contains 1,912 traces, or 31.87% coverage. Coverage is nearly flat across history length and positive grade, but declines sharply as the slate grows (Table C.3).

RFT reuses the shared evaluation prompt of Section B.2; it has no teacher prompt and no privileged label in the model input. Continuing the corresponding SFT adapter on the accepted responses is ordinary offline likelihood training: the objective is the SFT loss of Equation C.1 with the accepted set in place of the teacher set, where each accepted response $\hat{y}$ is the first of the eight samples from the frozen $\pi_{\mathrm{SFT}}(\cdot \mid x)$ that passes the verifier. The sampling step is on the frozen initializer, but the subsequent update is not online policy optimization. Table C.4 gives the complete settings and Table C.7 the development trajectories.

On development, no 1.7B RFT checkpoint exceeds the SFT initializer; the maximum-dev rule selects step 62. On test it reaches .223, compared with .225 for SFT, a paired difference of $-.002$ with 95% CI $[-.028, +.023]$. At 8B the development curve does exceed the initializer and selects step 90, but the selected checkpoint reaches .211 on test against .237 for SFT, a paired difference of $-.026$ with 95% CI $[-.047, -.005]$.

Setting	Value
Models	OpenOneRec 1.7B Pro and 8B Pro
Initialization / update	Base model / fresh LoRA $r = 16$, $\alpha = 32$, dropout 0, all linear layers
Teacher	Frozen merged SFT model at the same scale
Student and teacher input	Identical SID-only prompt and student-generated response prefix
Training data	Shared post-training prompt set, label-free, one prompt per user
Rollout	One response per prompt; temperature 1.0, top-p 1.0
Prompt / response / model caps	3,072 / 512 / 4,096 tokens
Distillation	Renormalized teacher top-32 forward-KL approximation
Loss aggregation	Response tokens only; sequence mean, then minibatch mean
Other learning signals	No task reward, policy-gradient loss, or reference KL
Teacher-query budget	6,000 student trajectories, exactly one per training prompt
Batch / horizon	30 / 200 updates, one pass over the prompts
Learning rate / warmup	5×10^{-5}, constant after 6 warmup steps
Checkpoints	Every 25 updates; all eight evaluated on 400 development users at each scale
Runtime	Two A100 GPUs; verl 0.8.0; PyTorch 2.10.0; vLLM 0.19.1

Table C.5: Frozen OPD specification. Its feedback and initialization differ from the other arms, while its task, source prompt keys, and evaluation benchmark remain shared.

C.3 On-Policy Distillation

OPD tests whether a task-adapted policy can provide dense supervision on the states visited by a Base student. At each scale, it uses a fresh LoRA on the Base model as student and the frozen, merged SFT model at the same scale as teacher. Both receive the exact SID-only prompt of Section B.2. The student samples the response, and at response position t the teacher receives the same prompt together with the student prefix $y_{<t}$. During OPD, neither policy is directly supplied with a caption, positive label, or answer hint, and no teacher-generated trajectory is used.

For state $s_t = (x, y_{<t})$, let $V_{32}(s_t)$ be the teacher's 32 highest-probability tokens. We renormalize the retained teacher mass,

$$\widetilde{p}_T(v \mid s_t) = \frac{p_T(v \mid s_t)}{\sum_{u \in V_{32}(s_t)} p_T(u \mid s_t)}, \qquad v \in V_{32}(s_t),$$

and minimize

$$L_t = \sum_{v \in V_{32}(s_t)} \widetilde{p}_T(v \mid s_t) \left[\log \widetilde{p}_T(v \mid s_t) - \log p_S(v \mid s_t)\right].$$

The student probability remains normalized over its full vocabulary. This is a computational top-32 approximation to forward distillation, not a task reward. Loss is applied only to sampled response tokens, averaged within each sequence, then across the minibatch. Task reward, policy-gradient loss, and a separate reference KL are disabled. Table C.5 gives the frozen specification and Table C.7 the development trajectories.

The prespecified maximum-dev rule selects step 75 at 1.7B and step 50 at 8B. On the test anchor, the selected 1.7B checkpoint reaches .195 against its Base initializer at .114, a paired gain of $+.082$ with 95% CI $[+.044, +.120]$; it is valid for all anchor users, with no identity permutation or stage-1 loop. The selected 8B checkpoint reaches .210 against Base at .168, a paired gain of $+.042$ with 95% CI $[+.001, +.083]$, while its $-.027$ difference from SFT has 95% CI $[-.060, +.007]$. It shows no identity permutation or stage-1 loop, and 395 of 400 outputs are valid after the fixed extraction fallback.

C.4 Reinforcement Learning with Verifiable Rewards

RLVR starts from the selected SFT policy and applies ordinary GRPO to a fresh LoRA adapter, trained on the shared post-training prompt set. The frozen benchmark-development split at $k = 64, N = 30$ supplies 400 monitoring examples but no policy-gradient updates. Both runs complete 100 updates, with checkpoints saved every 20 updates.

Setting	Value
Models	OpenOneRec 1.7B Pro and 8B Pro
Primary initialization	Corresponding dev-selected SFT checkpoint, merged into the base
Policy update	Fresh LoRA $r = 16$, $\alpha = 32$, all linear layers
Training / monitor data	Shared post-training prompt set / 400 benchmark-dev users
Rollouts per prompt	8
Sampling	Temperature 1.0, top-p .95
Prompt / response / model caps	6,144 / 2,048 / 8,192 tokens
Training duration	100 optimizer steps at each scale
Global prompt batch / PPO mini batch	60 / 30
PPO epochs	1
Learning rate / warmup	10^{-5} / .03
KL coefficient	.001 against the merged SFT initializer
Evaluated checkpoints	Steps 20, 40, 60, 80, 100 at each scale
Selection criterion	Highest official two-stage benchmark-dev NDCG@5 among completed RLVR checkpoints; earlier checkpoint breaks ties
Selected checkpoint	1.7B: step 60; 8B: step 100
Reward ablation	Binary top-one reward; SFT initialization; 1.7B selects step 60 and 8B selects step 80

Table C.6: GRPO RLVR settings. Both scales share the data, reward, rollout, and per-update optimization settings.

Let r be the returned rank of the one observed positive. The verifier uses the benchmark's exact single-positive NDCG@5 reward:

$$R(y) = \begin{cases} 1/\log_2(1+r), & \text{if } y \text{ is valid and } 1 \le r \le 5, \\ 0, & \text{otherwise.} \end{cases}$$

A valid trajectory must close its `</think>` block and then emit only one JSON array containing exactly ten distinct in-range integers. A positive outside the returned top ten, a rank below the NDCG cutoff, or any format failure receives zero. NDCG@10, Hit@1, identity, and parser fields are logged only as diagnostics. Identity receives no special penalty. No rationale content, rationale length, history-specific, or slate-specific shaping term is added.

GRPO turns this reward into a group-relative policy gradient. For each prompt x, the policy samples a group of $G{=}8$ responses, and each response's advantage is its reward standardized within the group,

$$\hat{A}_i = \frac{R(y_i) - \text{mean}\{R(y_j)\}_{j=1}^{G}}{\text{std}\{R(y_j)\}_{j=1}^{G}}, \tag{C.2}$$

so only within-group reward differences drive the update, with no learned value model. Every token of response y_i shares the advantage $\hat{A}_i$, and the update maximizes the clipped surrogate

$$J(\theta) = \mathbb{E}_{i,t}\Big[\min\big(\rho_{i,t}\hat{A}_i,\ \text{clip}(\rho_{i,t}, 1-\varepsilon, 1+\varepsilon)\,\hat{A}_i\big)\Big] - \beta\, D_{\text{KL}}\big(p_\theta \,\|\, p_{\text{SFT}}\big), \tag{C.3}$$

where the expectation averages over sampled response tokens, $\rho_{i,t}$ is the token-level probability ratio between the current policy and the policy that sampled the group, p_{SFT} is the merged SFT initializer, and $\beta{=}.001$ (Table C.6). The clip range and loss aggregation follow the verl 0.8.0 defaults recorded with the run configurations.

At both scales, a controlled reward ablation keeps the corresponding SFT initializer and replaces the graded reward with

$$R_{\text{top1}}(y) = \begin{cases} 1, & \text{if } y \text{ is valid and } r = 1, \\ 0, & \text{otherwise.} \end{cases}$$

All other data, rollout, optimizer, checkpoint, and development-selection settings are unchanged. RLVR reuses the shared SID prompt of Section B.2 and has no teacher prompt. Table C.6 gives the frozen contract and Table C.7 the development trajectories.

Arm	Scale	Init	Checkpoint NDCG@5 on development							
RFT (steps 31–122 / 30–120)	1.7B	.220	.217	**.217**	.209	.217				
	8B	.217	.222	.250	**.254**	.253				
OPD (steps 25–200)	1.7B	–	.129	.139	**.211**	.197	.206	.185	.203	.202
	8B	–	.167	**.266**	.227	.227	.246	.243	.228	.234
RLVR (steps 20–100)	1.7B	.218	.220	.223	**.233**	.233	.225			
	8B	.230	.230	.224	.244	.248	**.301**			
RLVR top-one (steps 20–100)	1.7B	.211	.211	.211	**.218**	.213	.196			
	8B	.214	.230	.230	.235	**.259**	.251			

Table C.7: Development-selection trajectories at the anchor setting. Checkpoints are listed in training order; the selected checkpoint is bold. RFT evaluates four checkpoints (1.7B: steps 31, 62, 93, 122; 8B: steps 30, 60, 90, 120), OPD eight (every 25 updates), and RLVR five (every 20 updates). For the 1.7B RFT row the bold checkpoint is the highest-scoring RFT checkpoint even though it does not exceed the initializer. Init is the initializer's score under the same protocol; OPD initializes from the Base model, whose anchor scores appear in Table 1.

On test, the selected 1.7B RLVR checkpoint reaches .231 against .225 for SFT, a paired difference of $+.006$ with 95% CI $[-.015, +.026]$. The selected 8B checkpoint reaches .277 against the SFT result at .237, a paired gain of $+.040$ with 95% CI $[+.006, +.076]$; relative to Base, its cumulative gain is $+.109$ with CI $[+.069, +.149]$. All reported RLVR outputs are valid, with no identity permutation or stage-1 loop.

The reward ablations are also fully valid. At 1.7B, the top-one policy reaches .232 on test, a $+.007$ difference from SFT with 95% CI $[-.014, +.028]$ and a $+.002$ difference from the graded-reward policy with CI $[-.018, +.022]$. At 8B, the top-one policy reaches .232, a $-.004$ difference from SFT with 95% CI $[-.025, +.016]$, while its $-.045$ difference from the graded-reward policy has CI $[-.077, -.013]$. Binary and graded rewards are therefore unresolved at 1.7B, while at 8B the graded reward's gain is resolved, highlighting the value of rank-sensitive partial credit.

D Additional Results

This section collects the results that support the main text but do not fit there: the remaining anchor metrics and paired contrasts, the rendering, token, and stratum diagnostics, the display-order, prompt-wording, and rationale-quality audits, and qualitative cases.

D.1 Anchor Metrics and Paired Contrasts

Table 1 already reports the full anchor metric suite for the calibration rankers, the text references, and the existing models under all three renderings, together with NDCG@5 and the interface diagnostics for every post-training arm (Table 2). Table D.1 therefore adds only what is missing there: the remaining ranking metrics for the post-training arms. Table D.2 gives the full set of paired contrasts. Beyond the initializer intervals of Table 2, only two kinds of added contrast resolve: the primary RLVR policies' gains over their Base models, at both scales, and the 8B graded-reward gain over the binary reward. Every added comparison against a text reference or across scales stays within noise. One further contrast compares the two RLVR rewards directly rather than through their shared initializer: the graded NDCG@5 reward finishes $+.045$ above binary top-one correctness at 8B (user-paired bootstrap CI $[+.013, +.077]$), while at 1.7B the two rewards are indistinguishable ($-.002$, CI $[-.022, +.018]$).

Method	Scale	Initializer	Supervision	N@5	N@10	R@5	R@10	MRR
SFT	1.7B Pro	Base	Teacher traces	.225	.290	.358	.560	.208
OPD	1.7B Pro	Base	Teacher distributions	.195	.242	.310	.455	.178
RFT	1.7B Pro	SFT	Filtered rollouts	.223	.290	.358	.568	.206
RLVR	1.7B Pro	SFT	NDCG@5 reward	.231	.298	.353	.563	.218
RLVR	1.7B Pro	SFT	Hit@1 reward	.232	.298	.358	.565	.217
SFT	8B Pro	Base	Teacher traces	.237	.302	.360	.565	.223
OPD	8B Pro	Base	Teacher distributions	.210	.264	.318	.485	.197
RFT	8B Pro	SFT	Filtered rollouts	.211	.274	.338	.535	.195
RLVR	8B Pro	SFT	NDCG@5 reward	.277	.347	.425	.640	.257
RLVR	8B Pro	SFT	Hit@1 reward	.232	.304	.353	.573	.222

Table D.1: Ranking metrics for the post-training arms at the anchor setting ($k = 64$, $N = 30$, display seed zero, 400 test users). N@j and R@j denote NDCG@j and Recall@j. Anchors for Random, Caption, the text references, and the existing models appear in Table 1. All rows are SID-only at inference. Every row is fully valid with no identity permutation or stage-1 loop, except that 8B OPD yields a valid ranking for 395 of 400 outputs. Supervision terms are defined in Section 5.1. Repeated RLVR rows are controlled reward ablations.

A	B	Mean A	Mean B	Δ(A-B)	95% paired CI
SFT 1.7B	Base 1.7B	.225	.114	+.111	$[+.071, +.151]$
SFT 8B	Base 8B	.237	.168	+.069	$[+.028, +.109]$
GPT-5.6-Terra text	Gemini text	.256	.247	+.009	$[-.015, +.033]$
GPT-5.6-Terra text	SFT 1.7B	.256	.225	+.031	$[-.007, +.068]$
SFT 1.7B	Gemini text	.225	.247	-.022	$[-.061, +.016]$
SFT 8B	Gemini text	.237	.247	-.011	$[-.049, +.026]$
SFT 8B	SFT 1.7B	.237	.225	+.012	$[-.020, +.043]$
OPD 1.7B	Base 1.7B	.195	.114	+.082	$[+.044, +.120]$
OPD 1.7B	SFT 1.7B	.195	.225	-.030	$[-.065, +.004]$
OPD 8B	Base 8B	.210	.168	+.042	$[+.001, +.083]$
OPD 8B	SFT 8B	.210	.237	-.027	$[-.060, +.007]$
RFT 1.7B	SFT 1.7B	.223	.225	-.002	$[-.028, +.023]$
RFT 8B	SFT 8B	.211	.237	-.026	$[-.047, -.005]$
RLVR 1.7B	SFT 1.7B	.231	.225	+.006	$[-.015, +.026]$
RLVR Top-1 1.7B	SFT 1.7B	.232	.225	+.007	$[-.014, +.028]$
RLVR Top-1 1.7B	RLVR 1.7B	.232	.231	+.002	$[-.018, +.022]$
RLVR 8B	SFT 8B	.277	.237	+.040	$[+.006, +.076]$
RLVR Top-1 8B	SFT 8B	.232	.237	-.004	$[-.025, +.016]$
RLVR Top-1 8B	RLVR 8B	.232	.277	-.045	$[-.077, -.013]$
RLVR 1.7B	Base 1.7B	.231	.114	+.117	$[+.075, +.159]$
RLVR 8B	Base 8B	.277	.168	+.109	$[+.069, +.149]$
RLVR 8B	Gemini text	.277	.247	+.029	$[-.006, +.065]$
RLVR 8B	GPT-5.6-Terra text	.277	.256	+.021	$[-.013, +.054]$

Table D.2: User-paired NDCG@5 comparisons at the anchor setting. Comparisons with text-view LLMs are diagnostic because model family and item representation both differ.

D.2 Rendering, Tokens, and Engagement Strata

The matched-view controls, which cross backbone scale with the three item-matched input views at the anchor setting, appear in Table 1. The only diagnostic not reported there is final two-stage validity: the SID view resolves every anchor example at both scales, and the text and combined views remain between 99.5% and 100%. Table D.3 reports the matched prompt lengths.

The OpenOneRec tokenizer is one common yardstick across renderings, not the billing count of any commercial API. Measured from the recorded API usage metadata, the mean anchor text prompt is 17,815 tokens for Gemini-3.6-Flash, 21,317 for GPT-5.6-Terra, and 31,209 for Claude-Sonnet-5, each at or above the 16,722 of Table D.3, so the compactness ratios of Section 5.2 are conservative under the providers' own tokenizers.

View	$k = 64$	$k = 128$	$k = 256$
SID	845	1,165	1,805
Text	16,722	27,899	50,412
Both	17,255	28,752	51,905

Table D.3: Mean input tokens under the OpenOneRec tokenizer at $N = 30$.

Scale / stratum	Base	SFT	Δ [95% CI]
1.7B / grade 2	.132	.206	$+.074$ $[+.018, +.131]$
1.7B / grade 3	.095	.244	$+.149$ $[+.093, +.204]$
8B / grade 2	.186	.223	$+.037$ $[-.018, +.092]$
8B / grade 3	.150	.250	$+.100$ $[+.040, +.160]$

Table D.4: Anchor NDCG@5 by positive-engagement stratum.

The balanced SFT source pool yields a near-balanced accepted set (5,994 grade-2 and 5,997 grade-3 examples). Table D.4 breaks the SFT gain down by stratum: at both scales it is larger for grade-3 positives, the explicit likes, follows, and forwards.

D.3 Display-Order and Prompt-Wording Stability

Two parallel audits rescore the Gemini-3.6 text reference at the anchor setting, one under new candidate display orders and one under new prompt wordings. Both are judged against the same baseline: rerunning an unchanged condition. Because the reference decodes at the provider-default temperature, an identical rerun already moves mean NDCG@5 by up to $.014$, and the per-user rankings of two identical runs agree at Kendall $\tau \approx .54$.

The display-order audit rescores the anchor setting under the two reserved display orders and leaves the reference essentially unchanged: NDCG@5 is .247, .234, and .245 across the three orders, and the frozen order differs from the permutation average by $+.005$ with paired 95% CI $[-.007, +.018]$. The prespecified switch to a permutation-averaged headline is therefore not triggered. In each of these orders, the top-ranked choice is statistically indistinguishable from uniform over the 30 input positions ($\chi^2(29) \leq 27.6$ against a 42.6 critical value), so scores reflect item content rather than display position. A fourth order, scored together with the wording audit, leaves the frozen order $+.0065$ above the four-order average. The order sensitivity of each SID system is reported separately through its identity-permutation rate.

The prompt-wording audit rescores the anchor setting under three natural rewordings of the frozen prompt's task statement. Each keeps the output contract, the history and candidate blocks, and the "candidate X" convention, so the audit varies phrasing rather than the task. No rewording moves the anchor score out of the rerun range: the paired differences from the frozen prompt are $+.009$, $-.011$, and $-.002$, every 95% CI straddles zero, and the ranking-contract satisfaction rate is unchanged at 399–400 of 400. The rescored runs and both audit reports are included in the released artifacts.

D.4 Rationale-Quality Judge

This audit compares rationale quality, not ranking quality, for the two strongest rationale-emitting systems at the anchor setting: the 8B RLVR (NDCG@5 reward) policy on the SID view and the Gemini-3.6-Flash text reference. Both rationales are the reasoning blocks the two systems emitted during their frozen anchor evaluations, taken verbatim; nothing is regenerated, because a regenerated rationale would describe a system the paper never reports.

Protocol. For 100 seeded test users, the judge (Claude-Opus-5, API model `claude-opus-5` served through Vertex, medium reasoning effort) receives the caption-rendered history and candidate slate, both anonymized rationales, and both emitted rankings, and returns one verdict (A, B, or tie) per aspect. It is instructed not to infer the observed positive, not to compare the rankings themselves, and not to reward length or breadth outside the coverage aspect. Every pair is judged twice, once in each presentation order. A verdict is decided only when the two orders agree; cross-

Aspect	Gemini	RLVR	Tie	Hesit.	Flip
Item-claim accuracy	**66**	0	21	12	1
User-model accuracy	**37**	0	28	35	0
Trade-off argument	0	**50**	16	34	0
Coverage (judged)	**100**	0	0	0	0
Overall	**97**	0	0	3	0

Table D.5: Paired verdicts over 100 users. Gemini and RLVR columns count pairs decided for that arm in both presentation orders; Hesit. counts cross-order hesitations between a win and a tie; Flip counts pairs where the decided direction reversed across orders. Every one-sided sign test rejects at $p < 10^{-10}$. The judged coverage row and the overall verdict are reported for completeness: coverage is also measured programmatically below, and the overall verdict tracks it.

order disagreements are reported separately and never split. Across the 786 non-tie verdicts, the judge chose the first-presented rationale 48.98% of the time. Order-consistent pairs contribute one first- and one second-presented choice by construction, so the informative subset is the 86 non-tie verdicts from pairs that split across orders: the first-presented rationale wins 35 of them (sign test p=.11), so presentation order carries no measurable preference. Item-level and user-level accuracy are deliberately separate aspects: naming the wrong item is the expected failure of an arm that has never read a caption, whereas the user portrait summarizes the history as a whole. Averaging them would hide this distinction.

Results. Table D.5 gives the counts behind Figure 3. In 500 paired aspect judgments the decided direction reversed once; every other cross-order disagreement is a hesitation between a win and a tie, so the per-aspect direction is stable even where hesitation is common. Two quantities are computed directly from the rationale text rather than judged, using the frozen parser, which reads each explicit candidate reference. Coverage is the share of the ten ranked candidates that the rationale names at least once: .971 for Gemini (9.7 of the ten) against .297 for RLVR (3.0). Fidelity is the per-user Kendall τ between the order of first mention and the emitted ranking, restricted to candidates that are both mentioned and ranked, and averaged over users with at least two such candidates, which holds for all 100 users of both arms: .983 for Gemini and 1.0 for RLVR. Unmentioned candidates are excluded rather than tied at the bottom, so that coverage does not leak into the fidelity number.

Judge validity. Two independent checks. First, coverage has a programmatic proxy, the count of named candidates, so the judged coverage row can itself be scored: the computed winner is Gemini for 98 of 100 users with 2 exact ties, and the judge agreed on all 98 with zero disagreements. Second, controlled corruption: for three users we re-ran the protocol after corrupting the Gemini rationale in three ways, swapping two candidate indices inside the text (falsifying item claims while preserving length, style, and the discussed candidate set), substituting another user's rationale, and truncating the rationale to its top-one justification. Aspects targeted by a corruption stopped being won by the corrupted arm in all eight applicable cases, and aspects that should be unaffected held in three of four; the single miss became a cross-order hesitation, not a reversed verdict. The index swap never moved coverage, as designed.

Arms compared. The audit pairs the strongest SID policy with the text reference, since the existing models rarely emit a rationale to compare: counting reasoning blocks that are nonempty and terminated, base-1.7B completes 4 of 400 anchor rationales and base-8B completes 2 of 400, against 399 of 400 for the post-trained arms and 396 of 400 for Gemini. The dominant base behaviors are an empty reasoning block that skips directly to the ranking and a verbatim transcription of prompt SID tokens that runs into the stage-1 token cap. Judge outputs, corrupted variants, and the control report are included in the released artifacts.

D.5 Qualitative Cases

Every case below is quoted from the frozen result files of the formal runs; Case 3 additionally quotes one frozen example from the released SFT training split. Nothing is regenerated or edited, and no smoke-test output appears. Each case names the anonymized user row of the released build, the

system, and the setting; every evaluation shown uses the frozen prompt. Responses and captions are Chinese. Excerpts appear here in English translation with whitespace compressed, SID tokens verbatim, and personal names replaced by bracketed placeholders; the untranslated originals are in the released files.

Case 1: identity collapse. User row 67158, 8B Base, SID view, anchor setting. Stage 1 terminates after 46 output tokens with an empty reasoning block and the literal identity ranking:

Case 1: the complete stage-1 response

```
<think>
</think>[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

The response is format-valid, so no fallback triggers, and the observed positive, at display position 29, is dropped entirely. This is the behavior Identity% counts: 82 of 400 anchor users for this model.

Case 2: reasoning loop. User row 1671, same run. Stage 1 opens with one planning sentence and then transcribes the prompt's history SID groups, in prompt order, until the 2,048-token stage-1 cap; the reasoning block never closes:

Case 2: stage-1 excerpt (translated; SID tokens verbatim)

```
<think> Okay, I now need to analyze the user's viewing history and
the candidate items and find the ten most relevant candidates.
First, I should look carefully at the user's viewing history and the
characteristics of each candidate item.
The videos the user watched include:
<|sid_begin|><s_a_2890><s_b_5369><s_c_3086><|sid_end|>,
<|sid_begin|><s_a_2180><s_b_1625><s_c_6870><|sid_end|>, [the
transcription continues to the token cap]
```

All 149 looping anchor users of the 8B Base stop at exactly this cap. Forced extraction then returns the identity list for this user, and the positive, at display position 26, is again lost.

Case 3: the teacher demonstration and the student's view. The first example of the released SFT training split (Section C.1), a 64-item history with a 20-candidate training slate. The caption-reading teacher grounds its picks in readable content:

Case 3: teacher rationale excerpt (translated)

```
The user's history shows a multi-dimensional distribution of
interests.  Over the long term, the user pays fairly stable attention
to rural daily life, traditional home cooking, and warm family
interaction; the recent history leans more toward fashion and
outfits, makeup transformations, celebrity and appearance content,
and emotionally resonant campus-life stories.  [...]  Candidate 3
shows rural life and kitchen cooking, fitting the user's
long-standing preference for pastoral food content.  [...]
```

The item-matched student instruction renders every one of the same items as an SID group alone; its first history item is <|sid_begin|><s_a_946><s_b_6523><s_c_444><|sid_end|>, and no caption text appears anywhere in it. The student imitates reasoning grounded in text it cannot read, which is why instance-level item claims are the expected failure of the elicited policies (Section D.4).

Case 4: what the elicited policy argues, and what it gets wrong. User row 88506, anchor setting: the selected 8B RLVR policy against the Gemini-3.6 text reference, a pair judged in Section D.4. RLVR returns the positive at rank 3; Gemini returns it at rank 8. The RLVR reasoning block spends its 192 tokens on exactly three candidates, and its narrative order matches its returned ranking:

Case 4: the two rationales on the same slate (translated excerpts)

```
8B RLVR: ''Candidate 27 tells the plot and emotional entanglements of
the novel 'Sisyphe Bookstore', highly matching the novel-recap and
emotional-story themes that appear repeatedly in the user's history.
[...]  Candidate 12 leans toward campus youth and selfies, and
candidate 8 toward internet hot topics and comedic banter [...]
slightly inferior to candidate 27 in thematic fit.''
True caption of candidate 27:  ''A detailed interpretation of the
characters' emotional entanglements and complex relationships in the
novel 'No Longer Devoted', finally revealing the story's ending.''
Gemini-3.6, same candidate:  ''Candidate 27:  fits the user's viewing
habit of emotional-novel interpretation and story entanglements.''
```

No novel of the invented name appears anywhere in the slate or the shown history. The interest claim itself is grounded: the shown window contains two novel-recap captions and two dozen captions on emotional themes. The interest is right and the item identity is wrong, while Gemini names all ten of its returned candidates with accurate one-line descriptions. The judge decided item claims for Gemini and the trade-off argument for RLVR on this pair: the split of Figure 3, visible in a single user.

Case 5: long-history improvement. User row 174437. The 441-item history shows a persistent interest in chat-record screenshots and emotional-confession videos (17 history captions are explicitly chat-record posts), and the observed positive is another such video. From the candidates alone (k=0), the RLVR policy misses it: the positive is not among the returned ten. At k=64 and k=256 the same policy holds it at rank 2. The Base model drifts instead of improving: rank 7 at k=0, out of the returned ten at both k=64 and k=256. The Gemini reference ranks the positive first at k=64 and demotes it to rank 7 at k=256, where its top pick switches to an outfit-change video of a celebrity who dominates one stretch of the added history. One user reproduces the three curves of Figure 2b.

Case 6: an unresolved invalid output. User row 174401, 1.7B Base on the text view, anchor setting. Stage 1 closes its reasoning block after 2,033 tokens with a plausible interest summary but never produces a ranking, and forced extraction then emits `[2, 3, 8, 1, 2, 3, 8, 1, 2, 3]`: four indices looped to length ten. Four distinct indices remain after deduplication, the contract requires ten, and the example scores zero. The observed positive, candidate 3, sits inside the loop. The zero records a broken interface, not a missed target, which is the separation the diagnostics are designed to keep.

E Artifacts and Compute

The anonymized code release is available at https://anonymous.4open.science/r/rerankid-code-8D2B. It contains deterministic benchmark and training-data builders, split manifests, per-user result files, parsers, reward code, prompt templates, available model revisions, environment specifications, and commands for all reported runs. Every formal result records the rendering, split, k, N, display seed, decoder, and selected checkpoint when applicable. For models served through an external provider, we record an immutable revision when the provider exposes one and otherwise record the provider model alias used for the call. Released data builds, adapters, and result files carry integrity manifests, and formal training inputs are verified against them before each run. The release is under the MIT License. Artifact publication excludes API credentials, account identifiers, private machine paths, smoke data, and raw user data that cannot be redistributed; the underlying interaction data remains governed by the OpenOneRec access terms.

Training and inference use bf16 on A100 GPUs; SFT, RFT, and RLVR train on four GPUs, and OPD uses two, one for the student and one for the frozen teacher. Exact software versions are recorded in each method's configuration and result metadata; OPD and RLVR use verl 0.8.0. Checkpoint selection uses benchmark development rather than test. Per-arm supervision budgets, retention counts, and API call accounting are recorded in the released configurations, data statistics, and result files, and wall-clock time is recorded for the external-API evaluation runs.

Each reported configuration is one training run whose selected checkpoint is evaluated once per frozen test condition. Decoding is greedy for SID evaluation, and the seeded Random baseline is fixed.