SAIL: SAILING BEYOND TEXT VIA BEHAVIOR-GROUNDED LLM SUMMARIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Ads recommendation systems must transfer signals from the organic content side to the ads side, but content items and ads sit in separate modeling spaces with no directly alignable ID representation and text that rarely shares words, so rich behavioral signals on the organic side are hard to reuse commercially. Existing approaches use either discrete Semantic IDs or text-derived representations such as pretrained embeddings and prompt-generated summaries. In our evaluated cross-domain setting, raw-text baselines and prompt-only LLM summaries exhibit a similar empirical *surface-representation ceiling* where performance saturates around a shared reference level. We propose SAIL, a two-stage post-training recipe that keeps the LLM's contribution in explicit natural language rather than opaque Semantic IDs: SFT distills a structured summary from a supply-grounded teacher, denoising raw text and initializing the schema with weak behavioral context; RL with a listwise retrieval reward then performs *retrieval alignment*, selecting and shaping the collaborative signals that make the summary useful for matching items to ads. Co-occurrence context is used only as training-time privileged information; at inference the summary is written from item text alone. On I2A-BENCH, SAIL raises AUC by $+\mathbf{2.53}$ points over a strong raw-text baseline, remains effective under an independent judge and transfers to a different evaluation protocol, and its largest performance drops occur when fields capturing commercial and audience-bridging signals are removed. The learned summary also transfers, without retraining, to a downstream LLM-based CTR task and improves macro-average AUC over seven objectives. Our *code* is available online.

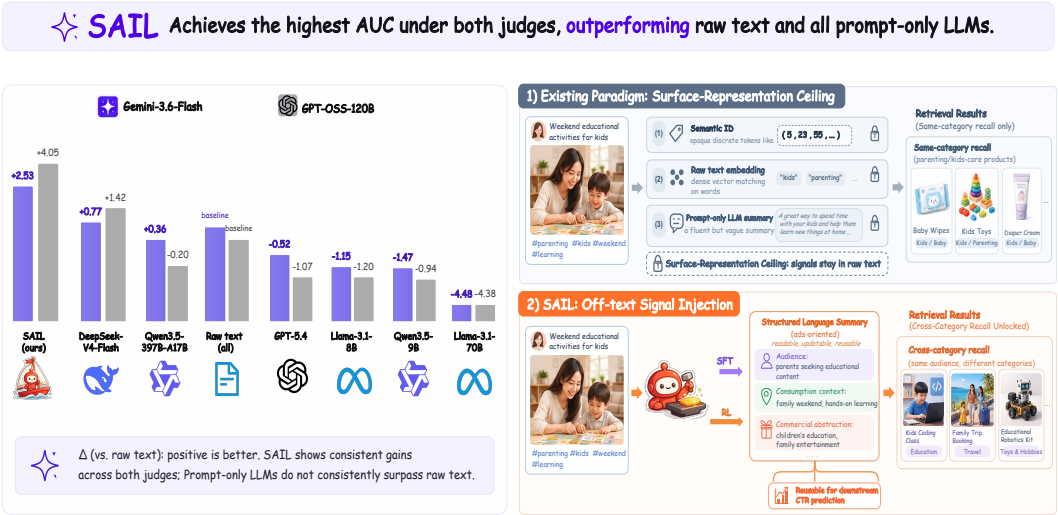

Figure 1: **SAIL achieves the highest AUC under two independent judges (left) by keeping the LLM's contribution in a readable structured summary (right).**

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.

1 INTRODUCTION

Ads recommendation systems aim to match ads to their intended users, and this matching depends on how well signals from the organic content side transfer to the ads side. On the organic side, users produce rich behavioral signals (views, likes, comments, saves) that traditional recommenders exploit through behavioral IDs and co-occurrence patterns (Linden et al., 2003; Kang & McAuley, 2018; Sun et al., 2019; He et al., 2020), with mature results when interactions are dense. Ads recommendation could leverage the same signals, but in practice content and ads sit in separate modeling spaces with no directly alignable ID representation, and their text rarely shares words. A parenting note, for example, may share both audience and consumption context with a coding-class ad for children, yet the note's text focuses on daily parenting anecdotes while the ad's text focuses on curriculum details, leaving no surface handle to connect the two sides. Rich behavioral signals on the organic side are therefore hard to reuse on the ads side, forming a persistent gap between them.

To bridge this gap, Large Language Models (LLMs) offer a way past the limits of ID-only recommendation, and have been used as feature enhancers, collaborative-embedding carriers, rankers, and query rewriters (Xi et al., 2024; Liu et al., 2023; Wei et al., 2024; Zhang et al., 2025a; Bao et al., 2023; Ma et al., 2023; Wang et al., 2023; Wu et al., 2024). The most active line is generative recommendation, which compresses each item into a discrete *Semantic ID* and trains a model to decode target IDs directly (Rajput et al., 2023; Deng et al., 2025), with recent variants adding reasoning over the ID sequence (Liu et al., 2025; OneRec Team, 2026). A parallel line stays in the dense-embedding regime, using pretrained encoders or prompt-generated LLM summaries followed by embedding retrieval (Huang et al., 2013; Zhou et al., 2018; Barkan & Koenigstein, 2016). These directions differ in machinery. Semantic IDs are constructed from item identity, interactions, and codebooks and follow their own paradigm; the other two, along with plain raw-text retrieval, share a structural property: *their representations are all derived from the raw item text or a compressed form of it*. On matching tasks that cross the organic-ads boundary, in our setting these text-derived approaches exhibit a similar empirical performance ceiling, which we call the **surface-representation ceiling**. From an information-theoretic view, the raw item text is a noisy observation of a latent *cross-domain collaborative signal*, the audience, consumption context, and commercial abstraction that connect an item to an ad; a text-grounded summarizer without external behavioral supervision is essentially a denoising step over the source text, and its retained information about the latent signal is limited by what the source text carries. In our evaluated setting, scaling the LLM, strengthening the encoder, or refining the prompt does not overcome this empirical ceiling, suggesting the limitation is not model capacity. A natural path forward is to expand that source by coupling the raw text with a cross-domain collaborative signal built from user behavior that directly links items to ads.

We therefore take a different route from Semantic-ID generative recommendation: rather than compressing items into opaque tokens, we *keep the LLM's contribution in explicit natural language* as a structured summary that a retriever and a downstream CTR model can read directly, and that can be inspected, updated, and reused across tasks. But explicit language alone is not enough; the language representation must be *post-trained* so that it is aligned to retrieval and carries collaborative signals shaped by user behavior. We propose SAIL (**S**ailing to **A**ds from **I**tems with **L**LM summaries), a two-stage recipe whose stages map onto the two claims above (Figure 1). *Stage one, SFT as denoising:* a Qwen3.5 397B teacher (Qwen Team, 2026) is distilled (Hinton et al., 2015) into a 9B student that writes summaries over a structured schema, compressing the raw text and approaching the ceiling. *Stage two, RL as retrieval alignment:* GRPO (Shao et al., 2024) with a listwise retrieval reward selects and shapes the audience, consumption context, and commercial abstraction signals in the summary so that they actually improve retrieval, moving beyond the ceiling. This is inspired by RL-based alignment of LLM outputs to downstream signals (Ouyang et al., 2022); here the signal is a listwise retrieval reward, and the target is off-text collaborative signal rather than helpfulness or reasoning. Our contributions are as follows:

- We introduce a new way to transfer training-time cross-domain behavioral signals into an inference-time text-only representation: rather than predicting an ID or ranking ads directly, SAIL learns an explicit language summary whose content is shaped by retrieval behavior. This representation is inspectable and reusable across retrieval and downstream tasks, and it encodes commercial signals that lie beyond the source text.
- We propose SAIL, a two-stage post-training recipe that keeps the LLM's contribution in explicit natural language rather than opaque Semantic IDs: SFT compresses raw text into

a structured schema while inheriting weak behavioral context from a supply-grounded teacher, and RL from a listwise retrieval reward then aligns the summary to retrieval, selecting and shaping the collaborative signals that actually improve matching.

- On I2A-BENCH, SAIL improves AUC by $+\mathbf{2.53}$ points over a strong raw-text baseline under a held-out judge, remains effective under an independent judge and transfers to a different evaluation protocol; the largest performance drops in a leave-one-field-out ablation occur when fields capturing commercial and audience-bridging signals are removed; the learned summary transfers, without retraining, to a downstream LLM-based CTR task.

2 PROBLEM FORMULATION

2.1 TASK SETUP AND THE SURFACE-REPRESENTATION CEILING

The item-to-ad (I2A) retrieval task ranks candidate ads from a large pool $\mathcal{A} = \{a_1, \ldots, a_N\}$ against a user-generated item x, where each ad is described by its own text. Since items and ads share the same text schema, we use *note* and *ad* interchangeably at the input level. Since items and ads on this platform carry no shared category taxonomy, we adopt an *embedding-dissimilarity criterion* as an operational proxy for cross-category matching: a positive pair (x, a) is labeled *cross-category-like* when the L2-normalized cosine similarity of x's and a's raw text falls below a small threshold (details in Appendix E). Under this criterion, 97.8% of positive pairs in our benchmark are embedding-dissimilar; they link an item to ads that share its audience and consumption scene rather than its words, and these links are not stated in the source text. For readability we simply write *cross-category* in later sections.

We consider retrieval functions $\mathcal{F}_{\text{raw}}$ operating on the raw text of items and ads (title, note content, caption, or their concatenations). We define the empirical surface-representation ceiling as the best AUC among the strongest raw-text settings we evaluate:

$$\text{AUC}_{\text{raw}} \;=\; \max_{f \in \mathcal{F}_{\text{raw}}^{\text{test}}} \text{AUC}(f, \mathcal{A}), \tag{1}$$

where $\mathcal{F}_{\text{raw}}^{\text{test}} \subset \mathcal{F}_{\text{raw}}$ is the set of raw-text baselines we test (Section 4.2). Beyond this empirical measurement lies an information-theoretic bound. The raw item text x can be viewed as a noisy observation of a latent cross-domain collaborative signal x^*, which captures the audience, consumption context, and commercial abstraction that connect an item to its target ads. Any text-grounded summarizer $s = f(x)$ without external behavioral supervision is essentially a denoising step over x: by the data processing inequality,

$$I(s; x^*) \;\le\; I(x; x^*). \tag{2}$$

The DPI motivates an information bottleneck: a deterministic summary over raw text alone cannot create new information about x^* beyond what x already carries. This does not by itself imply that AUC_{raw} (Eq. equation 1) is a theoretical upper bound: a better representation can reorganize existing information to be more usable by the downstream matcher. AUC_{raw} is only the best empirical performance among the raw-text baselines we evaluate, and moving beyond what can be inferred from raw text alone requires expanding the information source, coupling x with a signal y built from user behavior that links items to ads, which is the central goal of this work.

2.2 LISTWISE EVALUATION PROTOCOL

Relevance judgments over item-ad pairs are expensive to collect from humans at scale. We use a strong LLM as a *listwise judge* J. For a query q (either raw text or summary) and a candidate set $C = \{c_1, \ldots, c_K\}$ containing both positive and negative ads, the judge produces a permutation $\sigma \in S_K$ in a single pass, where $\sigma(i) < \sigma(j)$ indicates that c_i is judged more relevant than c_j. From σ we compute per-query AUC over all positive–negative pairs and NDCG@k over the top positions. The same judge serves both as the *evaluator* at test time and, in stage two, as the source of the *training reward* (Section 3.2).

2.3 STRUCTURED SUMMARIES

Rather than retrieving over raw text, SAIL rewrites each item into a structured summary and treats that summary as the item's intermediate representation. Because the summary is explicit natural

language, a downstream model such as a CTR predictor can read it directly. On the item side, a trainable summary policy π_θ (a Qwen3.5 9B LLM) rewrites each item into a summary $s \sim \pi_\theta(\cdot \mid x)$ that serves as the query for retrieval and judging. On the ad side, each ad is summarized once into the same schema and then kept fixed during training, so that any retrieval gain comes from a better item representation rather than a moving target.

3 METHODOLOGY

To break the surface-representation ceiling, the summary must carry commercial signals that the source text does not, and it must be written *for retrieval* rather than for readability. As the analysis in Section 1 shows, this cannot be reached by prompting a stronger LLM; the summary policy has to be trained against retrieval itself. SAIL does so in two stages (Figure 2): stage one (Section 3.1) is SFT as denoising: it distills a strong teacher into a summary policy π_θ (a Qwen3.5 9B LLM), so the policy inherits the output schema and basic commercial inference; stage two (Section 3.2) is RL as retrieval alignment: it applies GRPO with a reward from retrieval judgments so the policy can move past the teacher, selecting and shaping the signals in the summary that actually improve matching. Because the summary is explicit natural language, the same representation is also reusable by any downstream model.

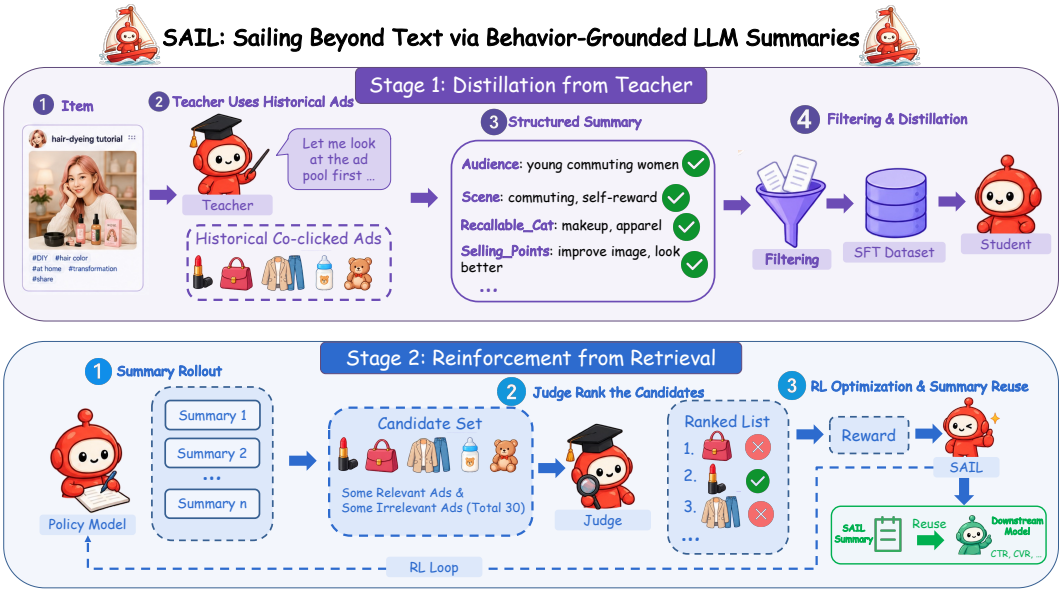

Figure 2: **SAIL distills a structured summary from a strong teacher and then aligns it to retrieval via reinforcement.** A summary policy rewrites each item into a structured summary that drives retrieval and judging: Stage 1 distills the schema and commercial inference from a 397B teacher into the 9B policy, and Stage 2 runs GRPO with a listwise-judge reward over a frozen ad pool, letting the policy discover inference patterns beyond the teacher.

3.1 STAGE 1: DISTILLATION FROM A TEACHER

Distillation is a starting point, not the source of our gains; it matters for two reasons. First, it fixes the representation *form*: the policy learns to emit the structured schema instead of free text, which makes the summary indexable and reusable (principle ❶). Second, it makes reinforcement feasible: running RL directly on the base model is impractical, since roughly 30% of its rollouts collapse into repetition loops that drown out the reward signal. What distillation *cannot* give is inference beyond the teacher; that is the job of stage two.

Teacher and grounding. We use Qwen3.5-397B-A17B, a mixture-of-experts model, as the teacher: it is the strongest model available to us, and its world knowledge covers the long tail of

products and scenes that a commercial inference layer must understand. For each training item x, the teacher writes a structured summary $\tilde{s}$ conforming to the schema in Table 5. Critically, the teacher prompt includes a small set of ads that have historically co-occurred with x in user sessions, so fields such as co-purchase scenes and adjacent content topics describe what the pool actually offers rather than free speculation; we give the teacher and judge prompts in Appendix C. This *supply-grounded prompting* prevents the teacher from hallucinating categories absent from the ad pool. These co-occurrence signals serve as training-time privileged information for teacher grounding only, and are not required at inference time: at test and deployment, the student policy π_θ takes an item text as its sole input.

Data filtering and training. We generate distillation data with JSON-mode decoding, guaranteeing that every teacher output parses without post-hoc repair, and keep only outputs whose fields are complete; the resulting distillation set $\mathcal{D}_{\text{sft}}$ is described in Appendix E. The student is a Qwen3.5 9B base model, trained with the standard negative log-likelihood objective:

$$\mathcal{L}_{\text{sft}}(\theta) \;=\; -\mathbb{E}_{(x,\tilde{s})\sim\mathcal{D}_{\text{sft}}} \sum_{t=1}^{|\tilde{s}|} \log \pi_\theta(\tilde{s}_t \mid x, \tilde{s}_{<t}). \tag{3}$$

Ad-pool summarization. The distilled student then summarizes the full ad pool into the same schema (Appendix E). These ad summaries are kept fixed during stage-two training, so every reward gain in RL reflects a real improvement in the item summary rather than a moving target; at evaluation time, ad summaries are regenerated by the same policy used on the item side, keeping the two sides symmetric.

3.2 Stage 2: Reinforcement from Retrieval

Although the teacher is exposed to behavioral context, SFT only teaches the student to imitate the teacher's outputs; it does not identify which parts of that transferred signal actually improve retrieval. A policy trained by SFT alone is capped by the teacher's inferences and does not yet write *for retrieval*. Stage two removes this cap by optimizing the summary directly against judged retrieval quality, so the policy can discover signals the teacher never produced, such as behavioral audience tags and consumption-scene triggers (principle ❷, ❸). Formally, we maximize

$$J(\theta) \;=\; \mathbb{E}_x \left[\mathbb{E}_{s\sim\pi_\theta(\cdot|x)} \left[r(s) \right] \right], \tag{4}$$

where $r(s)$ is the retrieval reward defined below. Since r is a black-box score from the judge rather than a differentiable function, we optimize it with GRPO (Shao et al., 2024), which fits this setting well: it is *critic-free* (each reward needs an expensive judge call, and dropping the critic halves that cost), its *group-relative advantage* matches our listwise reward (meaningful only across rollouts of the same item), and, unlike its original use with binary rewards, it works here with a *continuous, ranking-based* reward. For each item x, the current policy $\pi_{\theta_{\text{old}}}$ samples a group of $n = 8$ summaries, and following DAPO (Yu et al., 2025) we apply *dynamic sampling*: groups whose rollouts all receive the same reward yield zero advantage and are skipped.

Reward design. For each rollout s_i, the judge J ranks a candidate set of 30 ads (the item's positive ads plus sampled negatives) against s_i in one listwise pass (Section 2.2). Let $\mathcal{P}$ and $\mathcal{N}$ denote the positive and negative sets and $\text{rank}(c)$ the position of candidate c (rank 1 is best). The reward is a convex combination of AUC and NDCG@10 (Järvelin & Kekäläinen, 2002), capturing separation over the full list and quality at the top respectively:

$$r_i = \alpha\, \text{AUC}_i + (1-\alpha)\, \text{NDCG@10}_i, \qquad \alpha = 0.5, \tag{5}$$

$$\text{AUC}_i = \frac{1}{|\mathcal{P}||\mathcal{N}|} \sum_{p\in\mathcal{P}} \sum_{n\in\mathcal{N}} \mathbf{1}[\text{rank}(p) < \text{rank}(n)], \tag{6}$$

$$\text{NDCG@10}_i = \frac{\sum_{p\in\mathcal{P},\, \text{rank}(p)\le 10} 1/\log_2(\text{rank}(p)+1)}{\sum_{k=1}^{\min(|\mathcal{P}|,10)} 1/\log_2(k+1)}. \tag{7}$$

We arrived at this design after two failed alternatives: a pointwise rubric-score reward saturated near 0.94 for the base model and correlated only weakly with retrieval AUC ($\rho = 0.07$); an embedding-similarity reward saturated near 0.99 in the opposite direction. Both collapsed group-relative advantages, while the judge-based reward keeps a well-spread distribution throughout training.

Each rollout's candidate set contains the item's positive ads plus negatives sampled uniformly at random from the frozen ad pool, 30 candidates in total. We deliberately avoid hard negatives: by construction they are the ads a specific retriever confuses with the positive, so training against them pushes the policy to patch that retriever's mistakes rather than to identify what is truly relevant. Random negatives instead match the ad-pool distribution the policy faces at evaluation and deployment. In our early runs, RL with hard negatives quickly became unstable and failed to yield stable gains (Appendix F).

Optimization. Rewards within a group are normalized into a relative advantage $A_i = (r_i - \mu)/(\sigma + \epsilon)$, where μ and σ are the mean and standard deviation of the group. The policy is updated with the clipped surrogate objective inherited from PPO (Schulman et al., 2017):

$$\mathcal{L}_{\text{grpo}}(\theta) \;=\; -\mathbb{E}\left[\min\left(\rho_i A_i,\; \text{clip}(\rho_i, 1-\varepsilon, 1+\varepsilon) A_i\right)\right] + \beta\, D_{\text{KL}}\left(\pi_\theta \,\|\, \pi_{\text{ref}}\right), \tag{8}$$

where ρ_i is the sequence-level probability ratio between π_θ and $\pi_{\theta_{\text{old}}}$, and the KL term to the reference policy keeps updates stable. At deployment, the ad pool is summarized once offline and indexed while items are summarized on the fly with JSON-mode decoding; because the summary is plain text rather than a model-specific ID, the same representation can be reused by downstream models such as CTR (Section 4.6).

4 EXPERIMENTS

We evaluate SAIL on I2A-BENCH and ask five research questions: **RQ-1**: Does SAIL beat the raw-text baseline (Section 4.2)? **RQ-2**: How does each training stage contribute to the gain (Section 4.3)? **RQ-3**: Does training for retrieval hurt summary quality (Section 4.4)? **RQ-4**: Does the learned summary generalize to a different domain and evaluation protocol (Section 4.5)? **RQ-5**: Does the same summary transfer to a downstream CTR task without retraining (Section 4.6)? We first describe the benchmark and baselines (Section 4.1), then take up the five questions in turn.

4.1 SETUP

Benchmark. I2A-BENCH is built from a real-world platform snapshot: positive item–ad pairs are ads that received at least one click when co-shown with the item, aggregated over a 10-day serving window and split into training and held-out test days using an item-disjoint temporal split, so no test item appears during training. Because item and ad text are unlabeled free text with no shared taxonomy, we label a positive pair *cross-category* when its item- and ad-side embeddings have low cosine similarity; on the test set, 97.8% of positive pairs are cross-category, so the benchmark specifically stresses the setting raw text handles worst. For each test item we form a candidate set of 30 ads (mixing the item's positive ads with sampled negatives), a listwise judge ranks them in one pass, and the ad side is regenerated symmetrically by the same policy that produces the item side. During RL, ad summaries are precomputed and frozen to avoid a moving target; for final evaluation, each baseline is applied symmetrically to both sides. We report AUC as the primary metric because it summarizes the full ordering, and complement it with HR@k, NDCG@k, and MRR to show top-of-ranking behavior. Full details are in Appendix E and Appendix D.

Baselines. We compare two representation groups. *Raw-text* baselines retrieve using the item's raw text under four field settings: title only, caption only, note content only, and their concatenation; the strongest of these defines our empirical *surface-representation ceiling*, the best AUC among the evaluated raw-text settings. A dense retriever trained on click pairs is a natural alternative; however, our goal is to learn an explicit language representation that introduces commercially relevant signals not directly expressed in the source text. *LLM summary* methods rewrite each item into a summary: this includes prompting off-the-shelf LLMs without training: Qwen3.5 (9B and 397B-A17B) (Qwen Team, 2026), Llama-3.1 (8B and 70B) (Dubey et al., 2024), DeepSeek-V4-Flash (DeepSeek-AI, 2026), and GPT-5.4 (OpenAI, 2026), as well as our trained model SAIL. All methods share the same ad pool, judge, and candidate protocol and are applied symmetrically to both sides, so the observed differences arise from the resulting representations under the shared evaluation protocol; training and hyperparameter details for SAIL are in Appendix B.

Table 1: Main results. All values are absolute differences (Δ, in metric points) from the raw-text baseline (*title*+*content*+*caption*), whose value is omitted for confidentiality and denoted by −; higher is better. Per column, the best result is shaded in purple and the second-best in blue. SAIL numbers are averaged over 3 independent training seeds.

Method	Δ AUC		Δ HR@1	Δ HR@5	Δ HR@10	Δ NDCG@5	Δ NDCG@10	Δ MRR
	Gemini	gpt-oss						
Raw text								
Title Only	−15.02	−10.22	−11.50	−18.00	−9.50	−17.21	−14.51	−13.18
Caption Only	−6.14	−3.75	+2.50	−2.00	−3.50	−1.32	−1.52	+0.33
Content Only	−3.09	−0.57	+1.50	−2.00	−1.00	−2.33	−1.70	−0.31
Title + Content + Caption	–	–	–	–	–	–	–	–
LLM summary								
GPT-5.4	−0.52	−1.07	+3.00	−0.50	+0.50	+0.20	+0.65	+1.43
DeepSeek-V4-Flash	+0.77	+1.42	+3.50	+2.50	+1.00	+2.59	+1.85	+2.72
Qwen3.5-397B-A17B	+0.36	−0.20	+3.00	+1.50	−0.50	+1.63	+0.81	+2.14
Llama-3.1-70B-Instruct	−4.48	−4.38	−2.50	−2.00	−4.50	−3.65	−3.96	−2.58
Qwen3.5 9B	−1.47	−0.94	+4.00	+0.00	−2.50	+0.26	−0.15	+1.87
Llama-3.1-8B-Instruct	−1.15	−1.20	+3.00	−2.00	+1.00	−1.23	−0.09	+0.93
SAIL (ours)	+2.53	+4.05	+9.50	+7.00	+2.00	+7.52	+7.33	+7.80

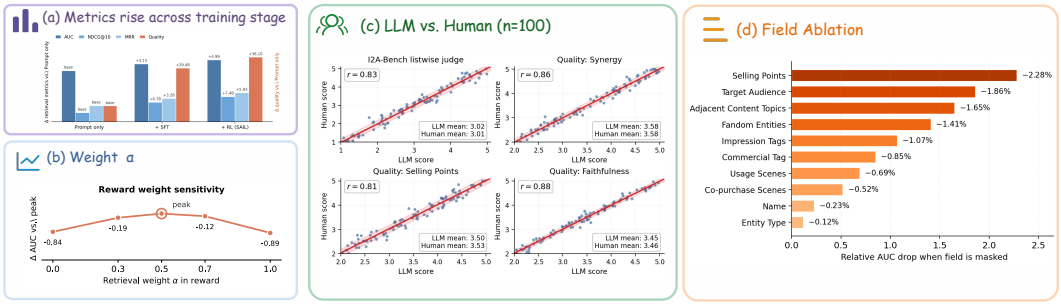

Figure 3: **Both stages help, the reward is stable at α=0.5, our LLM judges track human raters, and commercial/audience-bridging fields contribute most strongly to retrieval performance.** (a) Metrics rise monotonically across training stages. (b) AUC vs. α, peaking at 0.5. (c) LLM vs. 3-annotator human scores on 100 items; Pearson $r \in [0.81, 0.88]$. (d) Leave-one-out relative AUC drop when each schema field is masked from the SAIL summary.

4.2 MAIN RESULTS

(For RQ-1) SAIL beats the raw-text baseline on every metric and under both judges. We compare all methods under the listwise judgment protocol on I2A-BENCH. Table 1 reports the results under two independent judges. Three findings stand out. First, raw text is a stubborn baseline: concatenating title, content, and caption is stronger than any single field, and further gains from surface text saturate. Second, scaling the LLM does not, on its own, produce a summary that retrieves better than the raw text it summarizes: prompt-only variants across sizes and families sit close to the raw baseline, and larger models are not consistently better than smaller ones. Third, training the summary explicitly *for retrieval* does what scale cannot: SAIL remains the top-ranked method under two independent LLM judges (the training judge gpt-oss-120b (OpenAI, 2025) and a held-out Gemini 3.6 Flash (Google DeepMind, 2026) judge from a different provider, Appendix F), and its LLM-judge scores track a 100-item three-annotator human study (Section 4.4), so the ranking is not specific to the training judge. The learned summary also transfers to an Amazon cross-domain benchmark under a different evaluation protocol (Section 4.5). Within the I2A-BENCH evaluation, all methods use the same candidate protocol and evaluation procedure, so the observed difference is consistent with the resulting representations being the primary source of the gain.

4.3 ABLATION STUDIES

(For RQ-2) The two training stages play distinct and complementary roles. We decompose SAIL on the same Qwen3.5 9B model into prompting only, adding SFT, and adding RL on top; Fig-

ure 3 summarizes the ablations under the training judge (gpt-oss-120b (OpenAI, 2025)). Prompting alone falls *below* the surface-representation ceiling ($\Delta < 0$). SFT gains $+\mathbf{3.13}$ AUC over the base 9B and establishes the structured summary for the RL stage. RL on top of SFT adds another $+\mathbf{1.86}$ AUC and generalizes across both judges (Table 1); Figure 3(d) traces the gain to commercial and audience-bridging fields, and Appendix G illustrates this shift qualitatively on a single item.

4.4 SUMMARY QUALITY

Table 2: Summary quality across training stages on I2A-BENCH, scored by a five-aspect rubric.

Setting	Overall	Synergy	Audience	Selling	Scene	Faithful
Prompt only (base 9B)	12.37	11.60	8.87	13.74	12.09	15.53
+ Distillation (SFT)	41.85	41.48	42.80	42.03	42.69	40.23
+ Reinforcement (SAIL)	**50.47**	**48.58**	**50.07**	**50.81**	**50.95**	**51.95**

(For RQ-3) Training for retrieval does not sacrifice quality; quality improves alongside retrieval AUC. We score summaries offline with a *checklist-style* rubric along five aspects: synergy, audience, selling points, scene, and faithfulness. Figure 3(a) shows the overall quality score across training stages, and Table 2 reports the full per-aspect breakdown. Quality rises monotonically on all five aspects, faithfulness included, so training for retrieval does not drift the summary away from the item. Since training used *only* the retrieval reward, this is a by-product: writing for retrieval and writing a good commercial summary turn out to be aligned.

Table 3: Cross-domain generalization on the Amazon benchmark of Kim et al. (2026). SAIL is trained only on I2A-BENCH and transferred without retraining. Values for baselines other than the trained Semantic ID (SID; TIGER) model are quoted from Kim et al. (2026); the trained SID (Rajput et al., 2023) is our reproduction under the identical evaluation protocol. H@k is Hit@k and N@k is NDCG@k; higher is better.

Method	Beauty & Personal Care					Toys & Games				
	H@1	H@3	H@5	N@3	N@5	H@1	H@3	H@5	N@3	N@5
LightGCN	35.8	62.8	78.8	51.2	57.9	40.9	67.7	81.5	56.4	62.1
LightGCN++	38.3	65.2	80.2	53.4	59.7	**42.2**	68.5	82.5	57.3	63.1
SASRec	35.2	62.6	78.7	51.9	57.5	32.4	58.5	75.1	47.0	54.6
BERT4Rec	36.1	63.2	78.9	52.3	57.7	32.0	58.7	75.8	47.1	55.5
Zero-shot LLM	34.6	57.7	72.9	48.0	54.2	39.3	62.5	76.5	52.7	58.5
Trained SID (TIGER)	20.5	53.9	76.3	39.6	48.8	19.8	54.5	75.7	39.6	48.3
SAIL (ours)	**39.3**	**71.0**	**85.4**	**57.8**	**63.7**	41.1	**71.6**	**83.3**	**58.6**	**63.3**
Method	**Sports & Outdoors**					**Arts, Crafts & Sewing**				
	H@1	H@3	H@5	N@3	N@5	H@1	H@3	H@5	N@3	N@5
LightGCN	45.3	70.6	83.3	60.2	65.5	51.7	78.2	88.0	67.1	71.1
LightGCN++	**46.0**	70.8	84.0	60.5	65.9	**53.0**	79.9	88.8	**68.2**	**72.2**
SASRec	44.0	68.9	82.1	57.9	63.3	47.4	72.9	86.5	62.5	69.6
BERT4Rec	44.2	70.5	83.8	59.4	65.3	48.2	74.6	87.1	63.3	69.9
Zero-shot LLM	43.4	66.6	81.6	56.7	62.9	45.7	72.2	83.8	61.1	65.6
Trained SID (TIGER)	20.1	52.0	73.8	38.4	47.4	22.5	54.1	77.3	40.6	50.2
SAIL (ours)	45.7	**74.4**	**85.9**	**62.3**	**67.0**	49.7	**81.4**	**90.2**	68.0	71.7

4.5 CROSS-DOMAIN GENERALIZATION EVALUATIONS

(For RQ-4) The learned summary transfers zero-shot to a public cross-domain benchmark. To assess whether the learned representation generalizes beyond I2A-BENCH, we transfer SAIL, trained only on I2A-BENCH, to four Amazon subdomains (Beauty & Personal Care, Toys & Games, Sports & Outdoors, Arts, Crafts & Sewing) and compare against LightGCN (He et al., 2020), LightGCN++ (Lee et al., 2024), SASRec (Kang & McAuley, 2018), BERT4Rec (Sun et al., 2019), a zero-shot LLM prompt baseline, and a trained TIGER-style SID model (Rajput et al., 2023), following the evaluation setting of Kim et al. (2026). Table 3 shows that SAIL transfers to all four

subdomains and achieves the best score on 15 of 20 metrics, finishes second on four, and ranks third on one; the gain is especially clear on Beauty, where SAIL tops every metric, and on higher-k metrics across all subdomains, suggesting the learned summary produces useful ranking signals beyond the top-1 position even under a different task and evaluation protocol.

4.6 DOWNSTREAM TRANSFER

(For RQ-5) The summary transfers to a downstream CTR task unchanged, improving macro-average AUC on a held-out user bucket. We plug the SAIL summary into an LLM-based CTR model (Qwen3-0.6B) by replacing the raw item- and ad-side text with the SAIL summaries, keeping non-text features and training procedure fixed. Table 4 reports AUC across seven click targets: the summary lifts macro-average AUC by $+\mathbf{2.56}$ points, with the largest gains on lead-generation targets, message inquiry ($+\mathbf{9.42}$) and form submit ($+\mathbf{9.43}$), where capturing commercial intent from the note is central; engagement and mini-program targets improve modestly, while purchase and app activation drop by a few points, suggesting the transfer benefit concentrates on *intent-heavy* objectives while conversion-heavy targets rely more on user-side behavioral features.

Table 4: Downstream CTR prediction on a held-out user bucket. The CTR model and all non-textual inputs are held fixed; only the item- and ad-side text is swapped. Values are absolute AUC differences (Δ AUC, in points) from the raw-text baseline, whose value is omitted for confidentiality and denoted by $-$; higher is better.

Item/ad text	Like	Collect	Message	Form	Purchase	App Act.	Mini-prog.	Macro Avg
Raw text	$-$	$-$	$-$	$-$	$-$	$-$	$-$	$-$
SAIL summary	$+1.32$	$+1.85$	$+9.42$	$+9.43$	-2.34	-3.68	$+1.92$	$+2.56$

5 RELATED WORK

ID-based recommendation and text-based retrieval. Classical recommenders represent items by atomic IDs and learn associations from interaction history (Linden et al., 2003; Barkan & Koenigstein, 2016; He et al., 2020; Kang & McAuley, 2018; Sun et al., 2019), and text-based retrievers embed queries and documents into dense vectors using textual supervision or retrieval objectives (Huang et al., 2013; Karpukhin et al., 2020). Both work well when the match lies in co-occurrence or surface text, but degrade across the organic-ads gap.

LLMs for recommendation. LLMs have been introduced to supply the semantics and world knowledge that ID-based models lack, as feature enhancers, collaborative-signal carriers, rankers, and query rewriters (Xi et al., 2024; Liu et al., 2023; Wei et al., 2024; Zhang et al., 2025a; Bao et al., 2023; Ma et al., 2023; Wang et al., 2023; Wu et al., 2024). The most active line, generative recommendation (Geng et al., 2022), quantizes each item into a *Semantic ID* that a model decodes directly (Rajput et al., 2023; Deng et al., 2025; Liu et al., 2025; OneRec Team, 2026). The item is still encoded as opaque tokens tied to one generative model; OneRec Team (2026) introduce reasoning-oriented extensions for generative recommendation. Query- or document-rewriting approaches (Ma et al., 2023; Wang et al., 2023) typically train on generation likelihood. We instead use an explicit structured summary aligned to a listwise retrieval reward based on cross-domain behavioral relevance; this representation remains text-only at inference.

6 CONCLUSION

In this paper, instead of compressing an LLM's knowledge into Semantic IDs, SAIL keeps it in explicit natural language by distilling a structured summary from a teacher and aligning it with a listwise retrieval reward, surpassing the empirical ceiling of the evaluated baselines and transferring to CTR without retraining, a step toward inspectable and shareable intermediate representations.

AI USE STATEMENT

We used generative AI tools to polish the writing, tidy code, and assist with data anonymization scripts. All AI-assisted text and code were reviewed by the authors. The research questions, method design, experimental setup, and empirical claims are the authors' own. We take responsibility for the final content of this work.

ETHICS STATEMENT

I2A-BENCH is derived from anonymized internal commercial logs of a real-world content platform, uses only aggregated item–ad co-occurrence signals with no personally identifiable user information, and is not publicly released. For the Amazon evaluation we use publicly available benchmarks. Deploying such a model at scale requires standard safeguards around advertising fairness, transparency, and user consent, which are the responsibility of the deploying platform.

REPRODUCIBILITY STATEMENT

The SAIL training pipeline is described in Section 3, with implementation details in Appendix B, prompt templates in Appendix C, the summary schema in Appendix A, and dataset construction and evaluation protocols in Appendices E and D. The Amazon evaluation (Section 4.5) follows the public setting of Kim et al. (2026), and the TIGER-style SID baseline is reproduced under the identical protocol. Code is released in the anonymous repository at `https://anonymous.4open.science/r/SAIL-FD97`; because the underlying platform data is confidential, the repository ships with mock data files that only verify the code runs end-to-end, and business-sensitive absolute values are reported as Δ against a reference baseline ($-$ in the tables).

REFERENCES

Keqin Bao, Jizhi Zhang, Yang Li, Jie Wang, Fuli Feng, and Xingyi Niu. Tallrec: An effective and efficient tuning framework to align large language model with recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems*, pp. 1007–1014, 2023.

Oren Barkan and Noam Koenigstein. Item2vec: Neural item embedding for collaborative filtering. In *2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP)*, pp. 1–6, 2016.

DeepSeek-AI. Deepseek-v4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*, 2026.

Jiaxin Deng, Shiyao Wang, Kuo Cai, Lejian Ren, Qigen Hu, Weifeng Ding, Qiang Luo, and Guorui Zhou. Onerec: Unifying retrieve and rank with generative recommender and iterative preference alignment. *arXiv preprint arXiv:2502.18965*, 2025.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. Recommendation as language processing (rlp): A unified pretrain, personalized prompt & predict paradigm (p5). In *Proceedings of the 16th ACM Conference on Recommender Systems (RecSys)*, pp. 299–315, 2022.

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

Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, YongDong Zhang, and Meng Wang. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 639–648, 2020.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. Learning deep structured semantic models for web search using clickthrough data. In *CIKM*, 2013.

Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446, 2002.

Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. In *ICDM*, 2018.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6769–6781, 2020.

Sunwoo Kim et al. ItemRAG: Item-based retrieval-augmented generation for LLM-based recommendation. In *Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pp. 3884–3888, 2026.

Geon Lee, Kyungho Kim, and Kijung Shin. Revisiting LightGCN: Unexpected inflexibility, inconsistency, and a remedy towards improved recommendation. In *Proceedings of the 18th ACM Conference on Recommender Systems (RecSys)*, 2024.

Greg Linden, Brent Smith, and Jeremy York. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1):76–80, 2003.

Qijiong Liu, Nuo Chen, Tetsuya Sakai, and Xiao-Ming Wu. A first look at llm-powered generative news recommendation. *arXiv preprint arXiv:2305.06566*, 2023.

Zhanyu Liu, Shiyao Wang, Xingmei Wang, Rongzhou Zhang, Jiaxin Deng, et al. Onerec-think: In-text reasoning for generative recommendation. *arXiv preprint arXiv:2510.11639*, 2025.

Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. Query rewriting for retrieval-augmented large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 5303–5315, 2023.

OneRec Team. Onereason technical report. *arXiv preprint arXiv:2606.06260*, 2026.

OpenAI. gpt-oss-120b & gpt-oss-20b model card. OpenAI, aug 2025. URL `https://openai.com/index/gpt-oss-model-card/`.

OpenAI. GPT-5.4 thinking system card. OpenAI Technical Report, 2026. URL `https://openai.com/index/gpt-5-4-thinking-system-card/`.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:27730–27744, 2022.

Qwen Team. Qwen3.5: Towards native multimodal agents. `https://qwen.ai/blog?id=qwen3.5`, 2026.

Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan Hulikal, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Tran, Jonah Samost, et al. Recommender systems with generative retrieval. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

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. *arXiv preprint arXiv:2402.03300*, 2024.

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 *CIKM*, 2019.

Liang Wang, Nan Yang, and Furu Wei. Query2doc: Query expansion with large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 9414–9423, 2023.

Wei Wei, Xubin Ren, Jiabin Tang, Qinyong Wang, Lixin Su, Suqi Cheng, Junfeng Wang, Dawei Yin, and Chao Huang. Llmrec: Large language models with graph augmentation for recommendation. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pp. 806–815, 2024.

Likang Wu, Zhi Zheng, Zhaopeng Qiu, Hao Wang, Hongchao Gu, Tingjia Shen, Chuan Qin, Chen Zhu, Hengshu Zhu, Qi Liu, Hui Xiong, and Enhong Chen. A survey on large language models for recommendation. *World Wide Web*, 27(4):60, 2024.

Yunjia Xi, Weiwen Liu, Jianghao Lin, Xiaoling Cai, Hong Zhu, Jieming Zhu, Bo Chen, Ruiming Tang, Weinan Zhang, and Yong Yu. Towards open-world recommendation with knowledge augmentation from large language models. In *Proceedings of the 18th ACM Conference on Recommender Systems*, pp. 12–22, 2024. doi: 10.1145/3640457.3688104.

Qiying Yu, Zheng Sun, Yufei Shen, Jie Chen, Wei Ye, and Shikun Zhang. Dapo: An open-source llm reinforcement learning system with iterative self-improvement. *arXiv preprint arXiv:2503.14476*, 2025.

Yang Zhang, Fuli Feng, Jizhi Zhang, Keqin Bao, Qifan Wang, and Xiangnan He. Collm: Integrating collaborative embeddings into large language models for recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 37(5):2329–2340, 2025a. doi: 10.1109/TKDE.2025.3540912.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, et al. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025b.

Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1059–1068, 2018.

A Structured Summary Schema

Table 5 gives the full specification of the ten-field summary schema used throughout the paper. Each field captures one dimension of commercial inference, written as an explicit, indexable value rather than free text.

Table 5: Structured summary schema. Each field captures a dimension of commercial inference.

Field	Type	Description
Entity Type	string	4-way: Product / POI / APP / Content
Name	string	Category or subcategory (with brand/model if applicable)
Fandom Entities	list	Explicit fandom entities (celebrity, IP, game, etc.)
Commercial Tag	string	One-sentence commercial positioning
Selling Points	list	Concrete selling points
Target Audience	list	Demographic/behavioral profiles
Usage Scenes	list	Usage occasions or contexts
Co-purchase Scenes	list	Co-purchase or bundle scenarios
Impression Tags	list	Short bracketed tags
Adjacent Content Topics	list	Likely next-hop content topics for the same audience

B Implementation Details

Stage 1: Distillation. We use Qwen3.5-397B-A17B as the teacher and Qwen3.5 9B as the student. Teacher outputs are generated with JSON-mode decoding (temperature 0.7, top-p 0.9). The student

is fine-tuned with LoRA (rank 64, $\alpha = 128$, dropout 0.05) for 2 epochs on 8 H800-80G GPUs, with an effective batch size of 128 (per-device 1 $\times$ gradient accumulation 16 $\times$ 8 GPUs), a learning rate of 5×10^{-5}, and a cosine schedule with 3% warmup (≈ 3 warmup steps out of $\sim$92 total). Training takes approximately 1.5 hours.

Stage 2: GRPO. We run about 700 optimization steps and select the best checkpoint by validation reward, with group size $n = 8$ and clip parameter $\varepsilon = 0.2$ (asymmetric high-clip $\varepsilon_{\text{high}} = 0.28$ following DAPO (Yu et al., 2025)). The KL coefficient β is set to 1×10^{-3} (low-variance KL estimator). The reward balance is $\alpha = 0.5$ (equal weight on AUC and NDCG@10). Dynamic sampling (following DAPO) drops groups with zero reward variance. RL training uses 32 H800-80G GPUs (16 actor, 4 rollout, 4 reference, and 8 judge-service) in fully asynchronous mode with staleness 8, and takes approximately 12 hours. The reference policy is fixed to the SFT checkpoint from Stage 1.

C Prompt Templates

We give condensed versions of the two main prompts. The original prompts are written in Chinese for a Chinese-language platform; we translate and abridge them here for readability, dropping the platform-specific heuristics and in-context examples.

Teacher prompt (supply-grounded). The teacher is asked to read an item (title, note content, caption) together with a small set of co-occurring ads that share the item's search sessions, and to output the ten-field summary schema (Table 5) as a single JSON object. The co-occurring ads serve only as context for inferring audience and adjacent interests; the summary must describe the item itself, and fields such as monetization paths must stay grounded in the supplied signals rather than free speculation.

Teacher summarization prompt (condensed)

```
You are a commercialization analyst. Given an item note (title /
content / caption) and {k} co-occurring ads from the same search
sessions (with co-occurrence counts), extract a commercialization
summary for downstream item-to-ad matching.
The co-occurring ads indicate what users of this item also click and
where the item sits in the interest graph; use them to infer audience
and adjacent content, but do not copy their selling points onto the
item.
Output a single JSON object with exactly these keys: entity type,
name, fandom entities, commercial tag, selling points, target
audience, usage scenes, co-purchase scenes, impression tags, adjacent
content topics.
Rules: every key must appear; ground the content in the item text and
do not fabricate numbers, brands, or certifications; any of the three
item fields may be empty (especially the title), in which case rely on
the caption and co-occurring ads rather than refusing; output only the
JSON object, no markdown or explanation.
```

Listwise judge prompt. The judge receives one item summary and the summaries of $n = 30$ candidate ads (the item's positive ads plus sampled negatives) and ranks all candidates by commercial match; the ranks of the positive ads are used to compute AUC and NDCG@10 for the reward.

Listwise judge prompt (condensed)

```
You are a commercialization ad ranker. You are given the
commercialization summary of an item note and the summaries of {n}
candidate ads (numbered 1 to {n}).
Task: rank the candidate ads from most to least commercially relevant
to the note. Relevant dimensions include category, audience, usage
scene, co-purchase, lifestyle, and aesthetic style.
```

```
Output format:  a single line with {n} comma-separated candidate IDs
(a permutation of 1 to {n}), from most to least relevant, in the form
<x, y, z, ...>.  Output nothing else; no explanation, reasoning, or
markdown.
```

D Evaluation Protocol Details

Candidate construction. Each evaluation instance consists of one item and a candidate set of 30 ads: the item's click-positive ads (one or more) plus negatives sampled uniformly at random from the frozen ad pool. For each test item we deterministically sample the negative ads using an item-dependent random seed, and reuse the identical candidate set across all methods, so any difference between methods reflects the representation rather than the sampled negatives. We fix the set size at 30 to keep the judge's context length manageable while still spanning a realistic recall shortlist. We use uniform random negatives for both training and evaluation to keep the two distributions aligned; harder retrieval-based negatives destabilized training in our preliminary RL experiments (Appendix F).

Judge model. We use gpt-oss-120b (OpenAI, 2025) (model identifier `gpt-oss-120b`; official model card dated 2025-08-05), a 120B-parameter mixture-of-experts model with reasoning disabled at inference time, as the listwise judge for both reward computation (training) and final evaluation. The judge is kept frozen and is not updated during RL.

E I2A-Bench Dataset Statistics

I2A-Bench is constructed from a real-world platform snapshot. Table 6 summarizes key statistics.

Table 6: I2A-Bench dataset statistics.

Statistic	Value
Total item–ad pairs	∼100K
Training items (distillation)	∼6.6K
Training items (RL)	∼5.9K
Test items	200
Ad corpus (offline index)	∼13.6K
Test ad pool (per-query candidates drawn from)	∼2.7K
Cross-category pairs (%)	97.8%
Avg. item text length (tokens)	∼200
Avg. ad text length (tokens)	∼300
Item content categories	Open-vocabulary†
Ad product categories	Open-vocabulary†

† Raw notes/ads are unlabeled free text.

Cross-category definition. Because item and ad texts are unlabeled free text, we do not assume a fixed category taxonomy. Instead, for each positive item–ad pair we embed both sides' text (raw title, content, and caption concatenated) with Qwen3-Embedding-0.6B (Zhang et al., 2025b) and compute the L2-normalized cosine similarity; a pair is labeled *cross-category* when this similarity falls below 0.55, which corresponds to roughly the 97th percentile of the positive-pair similarity distribution on our test set. The cross-category fraction is not sensitive to the exact cutoff: it is 93.3% at 0.45, 96.3% at 0.50, 97.8% at 0.55, 98.6% at 0.60, and 99.1% at 0.65; a supermajority of positive pairs falls below any threshold in this range, confirming that item and target ad text are semantically dissimilar under the embedding. Under this criterion, 97.8% of positive pairs in the test set are cross-category, confirming that lexical and topical overlap between item and target ad is minimal and that retrieval must rely on semantic bridging rather than surface matching.

Positive label construction. Click-positive ads are defined as ads that received at least one click when shown alongside the corresponding item in the platform's online serving logs, within a 1-day co-click window; we treat click co-occurrence as a behavioral positive signal rather than a full

relevance ground truth. Positive labels inherit the platform's exposure policy since only served ads can be clicked; we treat this as a scope limitation. Co-click edges are aggregated into daily snapshots over a 10-day serving window, with the first 7 days used as the training pool for distillation and RL and the last 3 days as the held-out test set. Items in the test set are fully disjoint from all training items (item-disjoint temporal split), so the evaluation measures generalization to unseen items. All behavioral context used for teacher grounding (co-occurring ads and co-occurrence counts) is drawn only from within the training period, so no test-day behavior leaks into the training pipeline. An item may have one or more positive ads; items with no recorded clicks are excluded from the evaluation set.

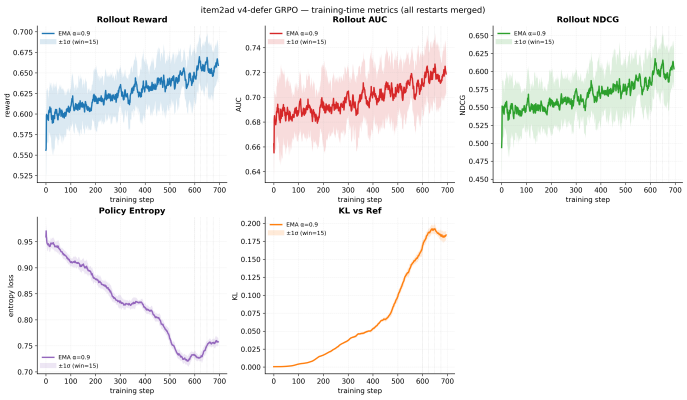

Figure 4: **RL steadily improves retrieval reward and rollout AUC after distillation.** Starting from the SFT checkpoint, rollout AUC rises steadily during training.

F Additional Results

Training curves. Figure 4 shows retrieval reward over the course of RL: reward rises steadily as GRPO optimizes against the listwise judge, and rollout AUC tracks it, showing consistent improvement in the training-time metric.

Hard-negative RL: instability. We also tried GRPO with hard negatives retrieved by a dense retriever (Section 3.2). Figure 5 shows the resulting collapse: reward drops after a brief rise, actor entropy collapses, and response length drifts to $\sim 4\times$ its starting value; we hypothesize that this instability is caused by the policy over-specializing to the retriever's error distribution.

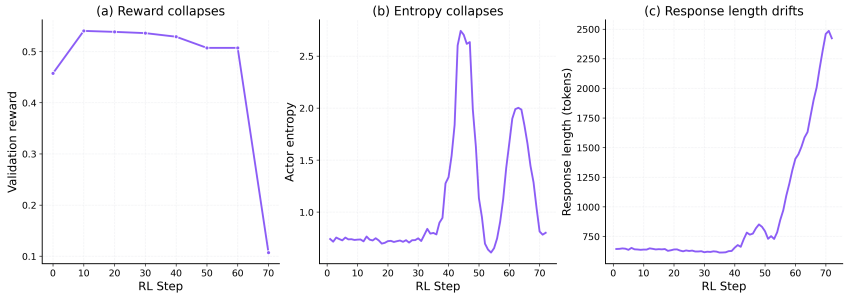

Figure 5: **RL with hard negatives collapses on all three training signals.** (a) validation reward drops sharply after a brief rise, (b) actor entropy collapses, and (c) response length drifts to roughly $4\times$ its starting value, indicating training instability.

Cross-judge robustness. We further verify that SAIL's advantage is not an artifact of our training judge (gpt-oss-120b) by re-evaluating all methods with a held-out judge from a different provider, Gemini 3.6 Flash (Google DeepMind, 2026) (model identifier `gemini-3.6-flash`; official

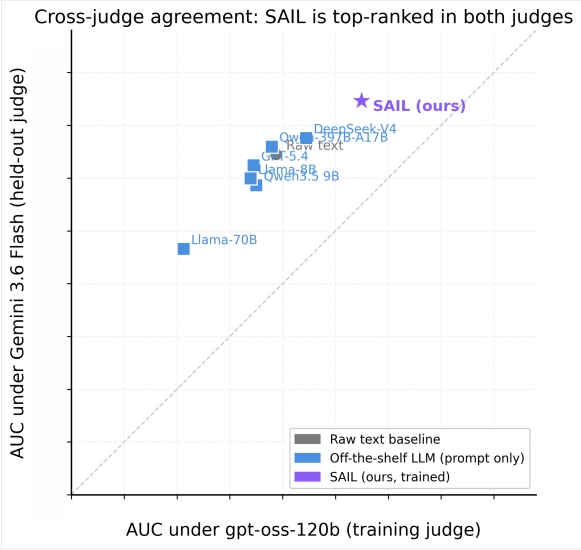

Figure 6: **SAIL ranks first under both judges.** Each point is one method; the $y = x$ diagonal marks judge agreement. Points show broad agreement across judges (Gemini is $\sim$5 pp higher), and SAIL (purple star) is at the top-right corner: highest AUC under both the training judge and the held-out judge.

model card dated 2026-07-21). All other setup is identical. Figure 6 plots each method's AUC under the two judges: the scores show broad agreement across judges, with SAIL highest and Llama-3.1-70B lowest under both.

Table 7: Ablation on the training stages, all on the same Qwen3.5 9B model, under the training judge (gpt-oss-120b). All values are absolute differences (in metric points) from the *Prompt only (base 9B)* setting, whose value is omitted for confidentiality and denoted by $-$; higher is better.

Setting	Δ AUC	Δ NDCG@10	Δ MRR	Δ Quality
Prompt only (base 9B)	–	–	–	–
+ Distillation (SFT)	+3.13	+4.78	+3.28	+29.48
+ Reinforcement (SAIL)	**+4.99**	**+7.48**	**+5.93**	**+38.10**

Training-stage ablation. Table 7 reports the exact retrieval metrics for the three settings on the same Qwen3.5 9B model with only the training procedure varying (prompt only, distillation, and reinforcement on top).

G QUALITATIVE ANALYSIS: CASE STUDY

SAIL replaces content-side tags with advertising-oriented abstractions. We trace a single item whose target ad is cross-category (a celebrity fitness clip vs. a phone endorsement, Figure 7). The connecting signal is not a shared topic but a shared audience-and-hotspot abstraction (*fan economy*), which is not explicitly stated in the item text. The base 9B summary reads the note literally and emits content-side tags (*fitness*, *sportswear*); the target ad ranks last (#30). Distillation adds bridge labels (*celebrity hotspot*, *fan economy*) but sometimes over-specifies a product category, so the target rises but is still ranked mid-pool. Reinforcement then drops the

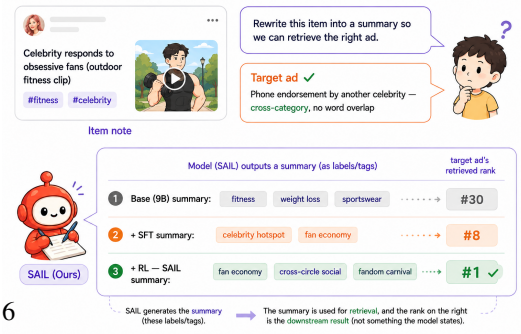

Figure 7: **Labels shift from content-side to**

misleading categories, re-anchors on the high-traffic entity, and sharpens the marketing wording, lifting the target to the top rank. This trace mirrors the field-level ablation in Section 4.3: the fields that drive retrieval, *target audience* and *adjacent content topics*, are precisely the ones that shift from surface tags to advertising-oriented abstractions across training stages.