

Beyond Query-Item Matching: Personalized Behavioral Relevance Modeling for E-commerce Search via Rubric-Guided Reinforcement Learning

Yang Liu^{1,*}, Chenxu Wang^{2,*}, Jianzhi Shao², Jun Ji¹, Xiaozhou Xu², Zhe Sun², Wei Hu^{1,†}

¹State Key Laboratory for Novel Software Technology, Nanjing University, China

²Alibaba Group, China

{yliu20.nju, junj.nju}@gmail.com, whu@nju.edu.cn

{carl.wcx, sjz250796, sz481403}@taobao.com, heixia.xxz@alibaba-inc.com

Abstract

E-commerce search systems typically separate query-item relevance matching from downstream personalized ranking. While effective for large-scale serving, this design leaves relevance levels dominated by objective query-item matching, forcing follow-up ranking to operate over candidate structures that may not reflect user behavioral preference. We define a new personalized behavioral relevance task, which predicts whether a relevance-stage candidate item is supported by the user’s behavioral evidence under current query intent. We develop a corresponding framework, BRIDGE, for Behavioral Relevance Inference with Dynamic Guidance from Evidence. It organizes multi-timescale behavioral context from online logs, obtains reliable supervision with multi-perspective annotation and adjudication, and trains an LLM to produce structured evidence-grounded decision rationales. It particularly incorporates sample-specific rubric rewards into GRPO to improve generated reasoning traces. Offline experiments validate that BRIDGE improves both prediction accuracy and reasoning quality. Online A/B tests show gains of 0.19% in active user-days and 0.67% in transaction-related user-days, both significant in statistics.

1 Introduction

E-commerce search is a central entry point that connects user shopping intents with products from large-scale inventories (Chen et al., 2025a, 2026). For users, effective shopping search requires both objective search relevance to the expressed query and alignment with personalized preference reflected in user behavior. In production systems, this balance is often achieved by a two-stage search pipeline: a *search relevance* model (Li et al., 2026) first assigns candidates to relevance score levels based on query-item matching, and a downstream

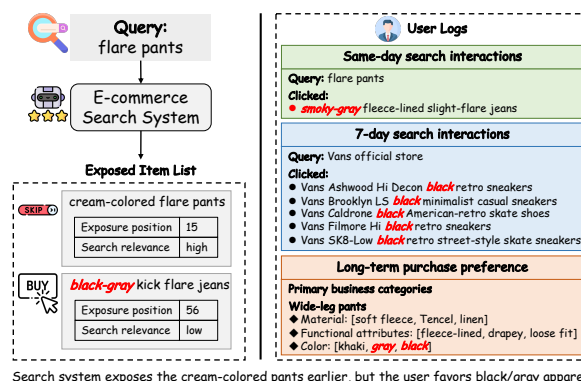


Figure 1: A motivating example.

personalized ranking model, often optimized for click-through rate (CTR) (Xiao et al., 2022), ranks them under the constraints of these levels.

While effective, this two-stage design is likely to create a mismatch between the two models. The search relevance model makes an upstream decision that strongly constrains item exposure, but its score levels are mainly determined by objective query-item matching. The personalized ranking model incorporates user signals only after this candidate structure has been formed. As a result, items that match a user’s behavioral preference may receive limited ranking opportunities if they are assigned to lower relevance levels, whereas textually matched but less preference-aligned items may remain in high-exposure levels. Fig. 1 depicts such a case: search relevance exposes a cream-colored item earlier, but user logs reveal a stronger behavioral preference for the black-gray item. A naive mitigation is to increase the number of candidates recalled for CTR prediction by including items on lower relevance levels. However, this enlarges the ranking candidate set and weakens the original role of relevance levels as explicit quality constraints.

Prior work (Chen et al., 2025b; Cao et al., 2026) has recognized this problem and addressed it from the *ranking side* by proposing relevance-CTR fu-

*Equal contributors

†Corresponding author

sion methods that jointly model relevance score levels, click probability, as well as user-specific relevance preference. User preference in these methods is represented by numerical features, semantic identifiers, or dense embeddings, and integrated with relevance and CTR signals into the final ranking scores. While effective for serving, such compact ranking representations cannot well utilize the rich evidence contained in user behavior, since they do not explicitly examine how behavioral preference signals correspond to fine-grained item attributes under current query intent.

In this paper, we tackle this problem from the *relevance-side* behavioral reasoning. Instead of injecting user preference into the final ranking scores, we introduce Personalized Behavioral Relevance (PBR) to assess whether a relevance-stage candidate item is supported by the user’s behavioral evidence under current query intent. To support this assessment, we collect multi-timescale behavioral context from historical logs and infer whether the item’s attributes align with the user’s preference under current search intent. This task formulation holds the existing two-stage search architecture: PBR is used as a relevance-level adjustment signal before CTR-oriented ranking, promoting behaviorally supported candidates so that the candidate structure better reflects both query-item matching and user preference alignment.

To instantiate this idea, we develop a new framework, BRIDGE, for Behavioral Relevance Inference with Dynamic Guidance from Evidence. BRIDGE takes user query, candidate item information, and user’s multi-timescale behavioral context as input, and returns a binary label indicating whether the candidate is supported by the user’s behavioral evidence. We employ an LLM to generate structured decision rationales that explicitly align query intent, item attributes, and behavioral evidence, benefiting from its deliberative reasoning capability. To obtain reliable label and rationale supervision, we construct training data through a multi-perspective annotation pipeline that combines heuristic behavior-derived priors, forward reasoning annotation, reverse verification, and judge-based adjudication. We further optimize BRIDGE with GRPO (Shao et al., 2024), into which we propose sample-specific rubrics as additional reward signals for generated outputs. This transforms relevance-side personalization to an evidence-grounded reasoning problem: BRIDGE combines multi-timescale behavioral context, ad-

judicated PBR supervision, structured reasoning, and sample-specific rubric rewards so that it is optimized not only for the final binary decision but also for how it uses behavioral evidence.

Our main contributions are outlined as follows:

- We introduce PBR, a relevance-side personalization task that estimates whether a candidate item is supported by the user’s behavioral evidence under the current query intent.
- We develop BRIDGE, a framework that learns structured evidence-grounded reasoning and uses sample-specific rubrics with GRPO to improve generated rationales. We also design a multi-perspective annotation and adjudication pipeline for reliable PBR supervision.
- Through offline experiments, we empirically show that BRIDGE improves the prediction accuracy of PBR. Furthermore, a real online A/B test demonstrates user engagement gains in production.

2 Related Work

2.1 Search Relevance Modeling

Search relevance modeling estimates whether a candidate item satisfies a user’s query intent. Conventional retrieval models, ranging from lexical matching such as TF-IDF (Ramos, 2003) and BM25 (Robertson and Zaragoza, 2009) to dense semantic matching based on pre-trained representations (Devlin et al., 2019; Khattab and Zaharia, 2020; Reimers and Gurevych, 2019), have established solid foundations for query-item matching in e-commerce search systems. Recent work (Tang et al., 2025; Dong et al., 2025; Wang et al., 2026a) re-frames search relevance as a reasoning problem with LLMs, enabling models to produce explicit rationales, handle complex or implicit query intent, and support structured relevance reasoning for production search. Follow-up work improves this paradigm with step-level reward assignment (Jiao et al., 2025), retrieval-augmented evidence use (Yang et al., 2026), multimodal information (Wang et al., 2026b), and deployable distillation (Xu et al., 2025). These studies focus on query-item relevance, and their goal is to determine if an item matches the query. Unlike them, our work defines PBR, which additionally conditions relevance-side decisions on the user’s behavioral context and refines relevance levels in terms of behavioral preference before downstream ranking.

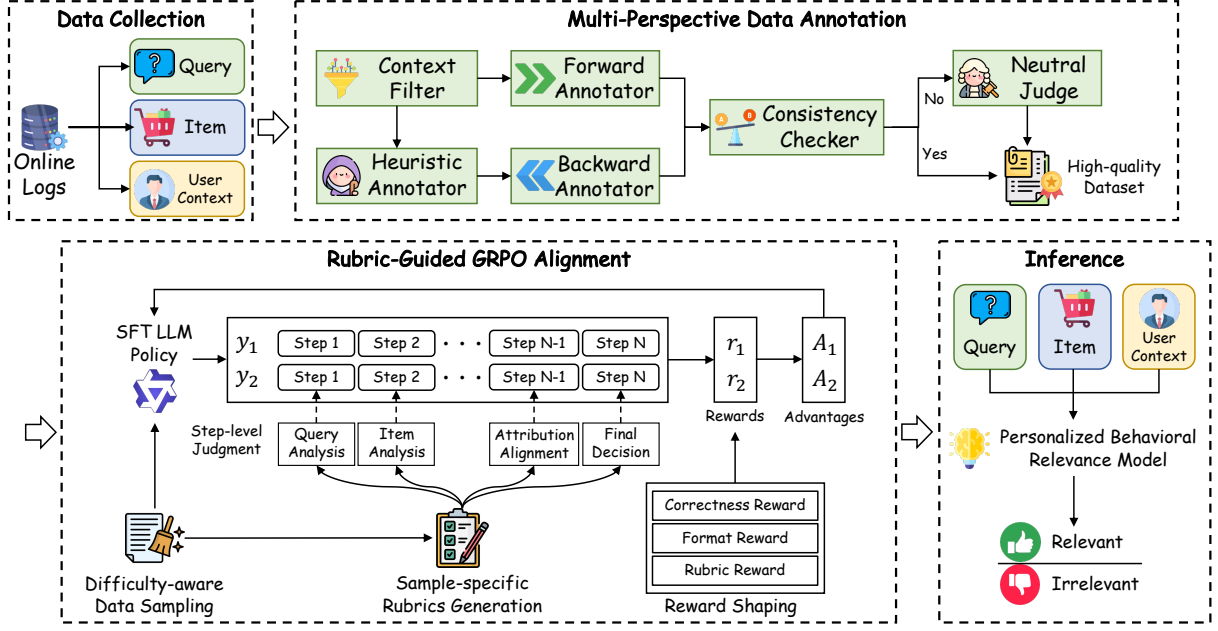


Figure 2: Overview of the BRIDGE framework.

2.2 CTR Prediction & Personalized Ranking

CTR prediction is a core component of personalized ranking in e-commerce search systems. Existing CTR models estimate post-exposure engagement through modeling sparse feature interactions (Juan et al., 2016; Guo et al., 2017; Lian et al., 2018), memorization-generalization trade-offs (Cheng et al., 2016), and user interests from behavior sequences (Zhou et al., 2018, 2019; Pi et al., 2020; Zhu et al., 2025). Closely related to our motivation, PRECTR (Chen et al., 2025b) observes that separately trained relevance and CTR models can produce inconsistent decisions, and thus proposes a unified relevance-CTR framework that models relevance score levels, click probability, and user-specific relevance preference. PRECTR-V2 (Cao et al., 2026) improves it with cross-user preference mining, exposure bias correction, and LLM-distilled encoder optimization. These studies mainly address the mismatch from the ranking side by improving the final ranking score, whereas our work defines PBR as an explicit relevance-side task and uses LLM reasoning to refine upstream relevance levels before personalized ranking.

3 Method

Fig. 2 illustrates the overall framework of BRIDGE. It first collects query-specific behavioral context from online logs (Sec. 3.2), then builds reliable supervision through multi-perspective annotation

and adjudication (Sec. 3.3), and finally trains the model with supervised reasoning (Sec. 3.4) and rubric-guided GRPO (Sec. 3.5) before deploying for inference.

3.1 Task Formulation

We formulate Personalized Behavioral Relevance (PBR) as a user-conditioned relevance-side prediction task. For a search query, let q denote the user query, i denote a candidate item, and H_u denote the behavioral context of user u . We explicitly decompose H_u into three temporal components:

$$H_u = (H_u^0, H_u^1, H_u^2), \quad (1)$$

where H_u^0 represents the real-time behavior from same-day search interactions, H_u^1 represents the short-term behavior within recent period, and H_u^2 represents the long-term preference evidence accumulated from historical behavior.

The goal is to predict a binary label

$$y = f_{\theta}(q, i, H_u^0, H_u^1, H_u^2), \quad y \in \{0, 1\}, \quad (2)$$

where $y = 1$ indicates that the candidate item is supported by the user’s behavioral preference under current query intent, and $y = 0$ otherwise. In an online system, this label is used to adjust the candidate’s relevance level before downstream personalized ranking: candidates predicted as $y = 1$ are promoted by one level. Thus, the adjusted level is no longer a purely objective query-item relevance

level, but a personalized relevance-side level that additionally incorporates behavioral preference.

Note that PBR does not replace conventional search relevance modeling or CTR prediction. Instead, it provides a relevance-side behavioral signal over relevance-stage candidates, so that downstream CTR-oriented ranking operates on relevance levels that better reflect both query-item matching and user preference alignment.

3.2 Multi-Timescale Behavioral Context

For each search query, we collect the behavioral context from online logs. The context provides behavioral evidence by describing the user’s recent and long-term behavioral patterns. We specifically consider three behavioral context categories:

- **Real-time search interactions** record the user’s same-day search interactions. Here, an interaction refers to the user’s feedback under a search query, such as clicks, add-to-cart actions, favorites, or purchases.
- **Short-term search interactions** aggregate the user’s search interactions in the past seven days, capturing short-term and recurring shopping intents beyond current session.
- **Long-term purchase preference** summarizes the user’s purchasing patterns in one year. It aggregates the user’s top price-zone preference records within each product category and category-property-value (CPV) preference records over items, together with a profile describing stable business characteristics.

These context categories complement each other: real-time interactions capture immediate intent, short-term interactions provide preference continuity, and long-term purchase preference describes more stable user needs. Because the behavioral context may contain many records unrelated to the current query, we treat each historical interaction or preference record as a retrieval entry and retain the top- K entries most similar to the current query within each context category. We use embedding similarity to filter historical behavior and construct a query-specific context, as detailed in Appendix A.

3.3 Multi-Perspective Data Annotation

Constructing reliable supervision is challenging since observed interactions may still be noisy,

while non-interacted items are not necessarily irrelevant because the user may have already selected other candidates. To resolve this, we construct labels through multi-perspective annotation, using repeated judgment and judge-based adjudication to improve annotation stability.

The pipeline consists of four stages. First, a heuristic annotator assigns each candidate a heuristic label using position-aware interaction rules, as detailed in Appendix B. This stage constructs a balanced annotation pool containing both likely-positive and likely-negative candidates. Note that these heuristic labels are used only as weak guidance rather than final labels. Second, a forward annotator independently reads the query, candidate item as well as behavioral context, and produces a structured reasoning trace and a binary label without observing the heuristic label. Third, a reverse annotator is given the heuristic label as a hint and verifies whether the evidence supports it, producing a verification rationale together with an independent binary label. Finally, when the forward and reverse labels disagree, a neutral judge inspects both rationales and decides whether the forward label should be accepted.

We only retain a sample if the forward label is corroborated by the reverse annotator, or if it is accepted by the judge during adjudication. We use the forward reasoning trace for supervised finetuning (SFT) as it matches the inference-time setting, where the model must make an independent decision without access to the heuristic label.

3.4 SFT with Structured Reasoning

We leverage the retained annotations to train the model with SFT. Each training instance consists of a query, a candidate item, the retrieved multi-timescale behavioral context, a forward reasoning trace, and a binary label. Rather than training the model to directly emit a label, we require it to generate a structured reasoning trace followed by the final binary decision. This design makes the prediction process evidence-grounded and aligns the training format with the annotation process.

The reasoning trace follows a fixed seven-step format: query intent analysis, item attribute analysis, real-time context analysis, short-term context analysis, long-term context analysis, attribute-level alignment, and final binary decision. This structure encourages the model to ground its decision in explicit query, item, and behavioral-context evidence, instead of relying on an unstructured

overall impression.

Given a training set $\mathcal{D}_{\text{SFT}} = \{(x, r, y)\}$, where x denotes the input, r denotes the forward reasoning trace, and y denotes the binary label, we optimize the model with the standard next-token prediction objective over the concatenated target sequence (r, y) . The resulting SFT model provides the initialization for the reinforcement learning (RL) stage.

3.5 Rubric-Guided GRPO

Prior work on relevance reasoning observes that a model may produce a correct label with an inconsistent or poorly grounded reasoning trace, and uses GRPO to mitigate this issue (Dong et al., 2025). However, outcome-only RL still provides only coarse final-label rewards and cannot directly evaluate whether the generated reasoning faithfully uses the required evidence. In this paper, we introduce sample-specific rubrics aligned with the structured reasoning trace as additional rewards for generated outputs during GRPO training.

For each input x , the current policy samples a group of candidate outputs $\{o_j\}_{j=1}^G$, where each output contains a structured reasoning trace and a binary decision. To make GRPO effective, we construct the RL pool with a difficulty-adaptive filtering strategy: the SFT model first generates multiple rollouts for each candidate sample, and we only retain samples whose rollout group contains both correct and incorrect predictions. This removes samples that are already too easy or too hard for the current policy, where group-relative advantages provide little useful learning signal.

The reward for each rollout checks output validity and combines the label and rubric rewards:

$$R(o) = \begin{cases} -1, & \text{invalid format} \\ \alpha R_{\text{label}}(o) + \beta R_{\text{rubric}}(o), & \text{otherwise} \end{cases}, \quad (3)$$

where $R_{\text{label}} \in \{0, 1\}$ is a binary reward indicating whether the final decision matches the label. $R_{\text{rubric}} \in [0, 1]$ measures the reasoning quality: an LLM judge evaluates each rollout against the sample-specific rubric checks, and the weighted pass rate is used as the reward. The format check requires the output to follow the structured reasoning format and produce a valid binary decision; invalid outputs receive a fixed reward of -1 without further reward computation. The key design is that rubric reward explicitly evaluates whether the generated reasoning uses the required evidence, rather

Splits	Examples	Positive proportion
SFT training set	30,000	56.7%
RL training set	18,325	57.3%
Test set	5,000	55.6%

Table 1: Dataset statistics.

than relying only on final-label correctness. Therefore, rubric-guided GRPO encourages the model not only to improve prediction performance, but also to produce reasoning traces that are better grounded in explicit behavioral evidence.

4 Experiments and Results

4.1 Experimental Setup

Dataset. We construct our dataset from online search logs collected over five days from 1688.com, a B2B e-commerce platform. The data are labeled by the multi-perspective annotation pipeline described in Sec. 3.3. The dataset contains 76,329 samples after annotation: 30K samples are used for SFT training, 5K samples are used for testing, and the remaining samples form the RL candidate pool. To construct the GRPO training set, we reuse the SFT model to generate eight rollouts for each sample in the RL candidate pool and retain samples whose rollout group contains both correct and incorrect predictions. This difficulty-adaptive filtering yields 18,325 samples for RL training. Table 1 summarizes the resulting data splits. All splits are performed by user ID to avoid possible user-level leakage across training and evaluation.

Baselines. We compare our BRIDGE with two categories of baselines: (1) Zero-shot LLM baselines include Qwen2.5-7B,¹ Qwen2.5-72B,² Qwen3-8B,³ and Qwen3-32B,⁴ which are prompted with the same input format without task-specific training. (2) The reranker baselines include Qwen3-Reranker-8B⁵ and its SFT variant. For compactness, they are denoted as Reranker and Reranker_{SFT} in the following table.

Metrics. Since PBR is formulated as a binary prediction task, we report accuracy, precision, recall, and F1 score on the test set. F1 is the primary metric because the task requires balancing preference-supported and unsupported candidates.

¹<https://huggingface.co/Qwen/Qwen2.5-7B-Instruct>

²<https://huggingface.co/Qwen/Qwen2.5-72B-Instruct>

³<https://huggingface.co/Qwen/Qwen3-8B>

⁴<https://huggingface.co/Qwen/Qwen3-32B>

⁵<https://huggingface.co/Qwen/Qwen3-Reranker-8B>

Models	Acc.	Prec.	Rec.	F1
Qwen2.5-7B	0.609	0.620	0.769	0.686
Qwen2.5-72B	0.641	0.643	0.796	0.711
Qwen2.5-7B + SFT	0.748	0.752	0.817	0.783
BRIDGE	0.798	0.780	0.888	0.830
Qwen3-8B	0.628	0.633	0.785	0.701
Qwen3-32B	0.708	0.699	0.835	0.761
Qwen3-8B + SFT	0.781	0.768	0.868	0.815
Reranker	0.495	0.556	0.447	0.496
Reranker + SFT	0.731	0.753	0.767	0.760
BRIDGE	0.806	0.791	0.885	0.835

Table 2: Offline evaluation results.

Implementation. For the behavioral-context retrieval, we employ Qwen3-8B-Embedding⁶ to compute query-entry similarity. We set $K = 10$ for both real-time and short-term search interactions, and $K = 5$ for industry preference records and CPV preference records. In the annotation pipeline, 50.6% of the samples receive consistent labels from the forward and reverse annotators, 27.7% are accepted by the judge after adjudication, and 21.7% are rejected because the judge disagrees with the forward label. The forward annotator, reverse annotator, adjudication judge, rubric generator, and rubric scorer use the same underlying LLM with separate role-specific prompts and inputs. To validate the resulting PBR labels, we further conduct a human validation on 200 randomly sampled test examples. Two PhD students independently annotate each example using the query, item information, and user behavioral context, and disagreements are resolved through discussion. The automatic labels achieve 93.5% agreement with the human labels, with a Pearson correlation of 0.8672 and Cohen’s κ of 0.8669, indicating strong agreement beyond chance. Both SFT and GRPO are trained on four H20 GPUs using the AdamW optimizer. For SFT, we initialize the model from Qwen3-8B and train it for three epochs with a learning rate of 1×10^{-5} . For GRPO, we train the model for two epochs with a learning rate of 1×10^{-6} and sample $G = 8$ rollouts for each input. We set the reward weights to $\alpha = 2$ for label correctness and $\beta = 1$ for rubric quality. Outputs with invalid format directly receive a reward of -1 .

4.2 Offline Evaluation

Table 2 reports the offline evaluation results. The zero-shot LLM baselines obtain relatively high recall but much lower precision, indicating that

⁶<https://huggingface.co/Qwen/Qwen3-Embedding-8B>

LLMs	Variants	Acc.	Prec.	Rec.	F1
Qwen2.5-7B	BRIDGE	0.798	0.780	0.888	0.830
	– GRPO	0.748	0.752	0.817	0.783
	– R_{rubric}	0.780	0.774	0.852	0.811
Qwen3-8B	BRIDGE	0.806	0.791	0.885	0.835
	– GRPO	0.781	0.768	0.868	0.815
	– R_{rubric}	0.792	0.779	0.873	0.823

Table 3: Ablation results.

prompting alone is likely to over-predict behaviorally supported candidates. Scaling up the backbones improves zero-shot performance, but these larger models still lag behind task-specifically trained models. SFT substantially improves the two Qwen backbones, raising F1 from 0.686 to 0.783 for Qwen2.5-7B and from 0.701 to 0.815 for Qwen3-8B. The reranker baseline performs poorly without task-specific training, mainly because our task differs substantially from conventional query-document retrieval, and its SFT variant improves but still remains below the generative SFT models. Our BRIDGE achieves the best overall performance, reaching an F1 of 0.830 with Qwen2.5-7B and 0.835 with Qwen3-8B. These results show that structured evidence-grounded reasoning and rubric-guided GRPO improve the relevance-side behavioral decision beyond prompting and supervised imitation.

4.3 Ablation Study

Table 3 verifies the contribution of the GRPO stage and the rubric reward. Removing GRPO consistently reduces F1 on both backbones, showing that SFT alone does not fully exploit the behavioral preference. The drop is more visible on Qwen2.5-7B, where F1 decreases from 0.830 to 0.783, suggesting that policy optimization is more useful for the weaker backbone. Removing the rubric reward also degrades all metrics on both backbones, indicating that sample-specific rubrics provide useful guidance beyond final-label rewards. Compared with the full model, the non-rubric variant preserves much of the prediction accuracy but loses part of the gain, which supports our design choice of using rubrics as auxiliary reasoning supervision rather than replacing label correctness.

4.4 Reasoning Quality Analysis

Because PBR is intended to produce evidence-grounded behavioral decisions, we evaluate the generated reasoning traces with the sample-

Steps	Qwen2.5	w/o GRPO	w/o R_{rubric}	BRIDGE
1	0.957	0.976	0.981	0.977
2	0.939	0.962	0.974	0.979
3	0.759	0.835	0.902	0.919
4	0.712	0.751	0.827	0.830
5	0.470	0.625	0.732	0.785
6	0.788	0.821	0.876	0.888
7	0.664	0.726	0.763	0.787

Table 4: Step-level reasoning quality evaluation with sample-specific rubrics.

specific rubrics. For each test instance, the rubric scorer checks whether the model output satisfies the seven reasoning-step criteria: query intent, item attributes, real-time context, short-term context, long-term context, attribute alignment, and final-decision grounding. We report the average pass rate for each step in Table 4. The first two steps, which analyze query intent and item attributes, obtain high scores across all models because they mainly involve local semantic understanding. The behavioral-context steps are more challenging, especially long-term preference analysis, where records are structurally complex and may include stale or weakly related evidence. BRIDGE improves these difficult steps most clearly, raising the long-term-context score from 0.625 after SFT to 0.785 after rubric-guided GRPO. It also improves final-decision grounding from 0.726 to 0.787, showing that the rubric reward encourages the model to justify the binary PBR decision with explicit behavioral evidence rather than only matching the final label.

4.5 Online A/B Test

For real-world evaluation, we deploy BRIDGE on 1688.com for an online A/B test, serving 5% of the main search traffic for three days. The online system contains four relevance levels, and BRIDGE is applied only to candidates in the top two levels. Items in the highest level already have strong query-item relevance, while items in the second-highest level are promoted to the highest level only when they are predicted as behaviorally supported. Each candidate can be promoted by at most one level before downstream ranking, preventing weakly relevant items from being moved directly to the highest exposure positions. We report active user-days and transaction-related user-days, which measure user-day instances with active search behavior and transaction-related conversion behavior, respectively. Compared to the current

system, BRIDGE raises active user-days by 0.187% ($p < 0.02$) and transaction-related user-days by 0.670% ($p < 0.01$), both statistically significant.

4.6 Deployment Efficiency and Scope

BRIDGE is deployed with offline inference and online caching, rather than synchronous online generation. Each inference sample is a user-query-item triple containing the query, one candidate item, and the user’s multi-timescale behavioral context. Using four H20 GPUs, the offline pipeline processes approximately 1.02 million samples per day, covering 20.9K queries from 10.8K users with an average of 48.6 candidate items per query. The average offline inference time is 4.12 seconds per query. During online serving, the search system only retrieves cached PBR decisions, and cache misses fall back to the original relevance pipeline, introducing negligible additional online latency.

BRIDGE is designed for high-value B2B buyers on 1688.com with sufficiently rich behavioral histories. For newly registered users or users with sparse interaction histories, the module is not applied and the original search relevance module is used. This deployment scope avoids unreliable preference inference under cold-start conditions and keeps the relevance-level adjustment restricted to cases where behavioral evidence is available.

5 Conclusion

This work revisits a basic assumption in industrial e-commerce search: relevance levels need not be purely objective query-item matching constraints. When user logs reveal stable behavioral preference, relevance-side decisions can also encode whether a candidate is suitable for the user under current query intent. From this perspective, BRIDGE complements ranking-side personalization by moving part of preference modeling upstream, before CTR-oriented ranking is applied. It also shows that LLM reasoning can be used not only to judge query-item relevance, but also to describe how query intent, item attributes, and user behavior are connected. We hope this work will encourage future search systems to treat relevance levels as preference-aware structures rather than fixed text matching outputs.

Acknowledgments

This work is supported by the CCF-1688 Yuanbao Cooperation Fund (No. CCF-Alibaba2025001).

Limitations

This work may have several limitations. First, the supervision construction relies on LLM-based annotators and judges. Although multi-perspective annotation and human label validation improve label reliability, the resulting labels, rationales, and rubrics may still inherit biases from the underlying LLMs or prompt designs. Second, our offline evaluation covers five days of logs and the online A/B test lasts three days. Longer offline and online evaluations are needed to establish temporal robustness. Third, our experiments are carried out on 1688.com and BRIDGE is currently applied only to users with sufficiently rich behavioral histories. Broader generalization requires cross-platform validation under different user behavior patterns and relevance-level mechanisms, as well as further study of sparse data settings.

References

- Shuzhi Cao, Rong Chen, Ailong He, Shuguang Han, and Jufeng Chen. 2026. PRECTR-V2: Unified relevance-CTR framework with cross-user preference mining, exposure bias correction, and LLM-distilled encoder optimization. *arXiv*, 2602.20676.
- Ben Chen, Xian Guo, Siyuan Wang, Zihan Liang, Yue Lv, Yufei Ma, Xinlong Xiao, Bowen Xue, Xuxin Zhang, Ying Yang, Huangyu Dai, Xing Xu, Tong Zhao, Mingcan Peng, Xiaoyang Zheng, Chao Wang, Qihang Zhao, Zhixin Zhai, Yang Zhao, and 9 others. 2025a. OneSearch: A preliminary exploration of the unified end-to-end generative framework for e-commerce search. *arXiv*, 2509.03236.
- Ben Chen, Siyuan Wang, Yufei Ma, Zihan Liang, Xuxin Zhang, Yue Lv, Ying Yang, Huangyu Dai, Lingtao Mao, Tong Zhao, Zhipeng Qian, Xinyu Sun, Zhixin Zhai, Yang Zhao, Bochao Liu, Jingshan Lv, Xiao Liang, Hui Kong, Jing Chen, and 4 others. 2026. OneSearch-V2: The latent reasoning enhanced self-distillation generative search framework. *arXiv*, 2603.24422.
- Rong Chen, Shuzhi Cao, Ailong He, Shuguang Han, and Jufeng Chen. 2025b. PRECTR: A synergistic framework for integrating personalized search relevance matching and CTR prediction. In *WWW*, pages 152–161.
- Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, Rohan Anil, Zakaria Haque, Lichan Hong, Vihan Jain, Xiaobing Liu, and Hemal Shah. 2016. Wide & deep learning for recommender systems. In *DLRS@RecSys*, pages 7–10.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, pages 4171–4186.
- Chenhe Dong, Shaowei Yao, Pengkun Jiao, Jianhui Yang, Yiming Jin, Zerui Huang, Xiaojiang Zhou, Dan Ou, and Haihong Tang. 2025. TaoSR1: The thinking model for e-commerce relevance search. *arXiv*, 2508.12365.
- Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: A factorization-machine based neural network for CTR prediction. In *IJCAI*, pages 1725–1731.
- Pengkun Jiao, Yiming Jin, Jianhui Yang, Chenhe Dong, Zerui Huang, Shaowei Yao, Xiaojiang Zhou, Dan Ou, and Haihong Tang. 2025. TaoSR-SHE: Stepwise hybrid examination reinforcement learning framework for e-commerce search relevance. *arXiv*, 2510.0797.
- Yu-Chin Juan, Yong Zhuang, Wei-Sheng Chin, and Chih-Jen Lin. 2016. Field-aware factorization machines for CTR prediction. In *RecSys*, pages 43–50.
- Omar Khattab and Matei Zaharia. 2020. ColBERT: Efficient and effective passage search via contextualized late interaction over BERT. In *SIGIR*, pages 39–48.
- Yupeng Li, Ben Chen, Mingyue Cheng, Zhiding Liu, Xuxin Zhang, Chenyi Lei, and Wenwu Ou. 2026. KuaiSearch: A large-scale e-commerce search dataset for recall, ranking, and relevance. *arXiv*, 2602.11518.
- Jianxun Lian, Xiaohuan Zhou, Fuzheng Zhang, Zhongxia Chen, Xing Xie, and Guangzhong Sun. 2018. xDeepFM: Combining explicit and implicit feature interactions for recommender systems. In *SIGKDD*, pages 1754–1763.
- Qi Pi, Guorui Zhou, Yujing Zhang, Zhe Wang, Lejian Ren, Ying Fan, Xiaoqiang Zhu, and Kun Gai. 2020. Search-based user interest modeling with lifelong sequential behavior data for click-through rate prediction. In *CIKM*, pages 2685–2692.
- Juan Ramos. 2003. Using TF-IDF to determine word relevance in document queries. In *ICML*, pages 29–48.
- Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using siamese BERT-networks. In *EMNLP*, pages 3980–3990.
- Stephen E. Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: BM25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv*, 2402.03300.

Tian Tang, Zhixing Tian, Zhenyu Zhu, Chenyang Wang, Haiqing Hu, Guoyu Tang, Lin Liu, and Sulong Xu. 2025. LREF: A novel LLM-based relevance framework for e-commerce search. In *WWW*, pages 468–475.

Chenxu Wang, Jianzhi Shao, Chi Zhang, and Tao Zhang. 2026a. STAR: Staged training with aligned reinforcement learning and multi-faceted distillation for interpretable e-commerce relevance. In *SIGIR*, pages 1–5.

Chenxu Wang, Chi Zhang, Fang-Yi Liang, Jianzhi Shao, Manyi Wang, and Tao Zhang. 2026b. TRACE: Term-level reasoning and chain-of-thought enhanced distillation for e-commerce multi-modal relevance learning. In *SIGIR*, pages 1–5.

Yunpeng Xiao, WeiKang He, Yu Zhu, and Jianghu Zhu. 2022. A click-through rate model of e-commerce based on user interest and temporal behavior. *Expert Syst. Appl.*, 207.

Tingqiao Xu, Shaowei Yao, Chenhe Dong, Yiming Jin, Zerui Huang, Dan Ou, and Haihong Tang. 2025. DyKnow-RAG: Dynamic knowledge utilization reinforcement framework for noisy retrieval-augmented generation in e-commerce search relevance. *arXiv*, 2510.11122.

Jianhui Yang, Yiming Jin, Pengkun Jiao, Chenhe Dong, Zerui Huang, Shaowei Yao, Xiaojiang Zhou, Dan Ou, and Haihong Tang. 2026. TaoSR-AGRL: Adaptive guided reinforcement learning framework for e-commerce search relevance. In *WWW*, pages 7955–7966.

Guorui Zhou, Na Mou, Ying Fan, Qi Pi, Weijie Bian, Chang Zhou, Xiaoqiang Zhu, and Kun Gai. 2019. Deep interest evolution network for click-through rate prediction. In *AAAI*, pages 5941–5948.

Guorui Zhou, Xiaoqiang Zhu, Chengru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep interest network for click-through rate prediction. In *SIGKDD*, pages 1059–1068.

Nana Zhu, Zixian Feng, Hang Wang, Xing Gao, Xinyi Wang, and Yuanxing Liu. 2025. Clarifying user preference with maximum entropy based recommendation. *Data Intelligence*, 7(2):527–548.

A Behavioral Context Filtering

The behavioral context extracted from online logs may contain many entries that are unrelated to the current query. Therefore, we apply query-aware filtering to the structured behavioral fields before constructing the final context.

For real-time and short-term search interactions, each entry is organized around a query and contains the items interacted with under that query.

For long-term purchase preference, entries correspond to structured records such as category-level price-zone preference records and category-property-value preference records. The stable business profile is kept as a whole and is not decomposed or filtered.

For each decomposed entry e , we compute its embedding similarity to the current query:

$$s(q, e) = \cos(\phi(q), \phi(e)), \quad (4)$$

where $\phi(\cdot)$ denotes the embedding encoder. Within each context category, we retain the top- K entries with the highest similarity scores and discard the remaining entries. The retained entries are then merged back into their original field format, so that the model receives a compact query-specific context while preserving the structure of the online-log records.

B Heuristic Annotation

The goal of heuristic annotation is not to produce labels, but to retrieve informative candidates from online logs and provide weak priors for the later annotation pipeline.

Position-aware priors. For each search exposure list, we estimate the user’s viewed range from observed interactions, and define $p_{\min}$ and $p_{\max}$ as the minimum and maximum positions among interacted items, respectively. Candidates with interactions are treated as likely positive cases. For non-interacted candidates, those ranked before $p_{\min}$ are treated as informative negatives; those ranked within $[p_{\min}, p_{\max}]$ are treated as possible negatives; candidates after $p_{\max}$ are not sampled because the user may not have reached them. These priors are used only to build a balanced annotation pool and to provide a weak hint to the reverse annotator.

C Prompt Templates

Inference prompt is shared by forward annotation, training, and inference to keep the input-output format consistent.

Inference Prompt

ROLE

You are an expert user behavior analyst for 1688.com, a B2B e-commerce search platform. Your task is to judge whether a candidate item is supported by the user’s behavioral evidence under the current search query.

TASK DEFINITION

The task is not conventional query-item textual relevance. Given a query, a candidate item, and the user's multi-timescale behavioral context, judge whether the item is supported by the user's behavioral preference under the current query intent. Infer whether the item matches the user's behavioral preference at the attribute level. If there is insufficient supporting evidence, output label 0.

QUERY
{query}

ITEM
Title: {item_title}
Other attributes: {item_attributes}

MULTI-TIMESCALE BEHAVIORAL CONTEXT
Real-time search interactions:
{real_time_context}

Short-term search interactions:
{short_term_context}

Long-term purchase preference:
{long_term_context}

OUTPUT CONSTRAINTS
Produce a structured seven-step reasoning trace:
1. Query intent analysis
2. Item attribute analysis
3. Real-time context analysis
4. Short-term context analysis
5. Long-term context analysis
6. Attribute-level alignment
7. Final decision rationale

Then output the final PBR label as exactly one of:
\boxed{0}
\boxed{1}

Label 1 means the item is supported by behavioral preference. Label 0 means the evidence is insufficient or does not support behavioral relevance.

Reverse annotation prompt follows the same structured format but additionally conditions on a heuristic label and prior-specific annotation notes. These notes describe how the heuristic label should be interpreted. Interaction-supported candidates are likely positives, but the annotator is reminded that interaction signals can still be noisy. Non-interacted candidates before $p_{\min}$ are likely negatives because they appear before the earliest observed interaction, making it likely that the user examined them sequentially and did not select them. Non-interacted candidates within $[p_{\min}, p_{\max}]$ are possible negatives because they fall inside the estimated viewed range, but the evidence that the user examined each of them is weaker than for candidates before $p_{\min}$.

Reverse Annotation Prompt

ROLE

You are an expert reviewer for user behavior analysis in e-commerce search. Your task is to verify whether a candidate item is supported by the user's behavioral evidence under the current search query.

TASK DEFINITION

The task is to judge whether the candidate item is supported by the user's behavioral preference under the current query intent. The heuristic prior below is only a weak hint. You should check whether the evidence supports it or whether it should be overturned.

QUERY
{query}

ITEM
Title: {item_title}
Other attributes: {item_attributes}

MULTI-TIMESCALE BEHAVIORAL CONTEXT
Real-time search interactions:
{real_time_context}

Short-term search interactions:
{short_term_context}

Long-term purchase preference:
{long_term_context}

HEURISTIC PRIOR
Heuristic label:
{heuristic_label}

Prior-specific annotation notes:
{prior_specific_notes}

OUTPUT CONSTRAINTS
Produce a structured verification trace with the following steps:
1. Query intent analysis
2. Item attribute analysis
3. Real-time context analysis
4. Short-term context analysis
5. Long-term context analysis
6. Prior-aware evidence verification
7. Final decision rationale

Then output the final PBR label as exactly one of:
\boxed{0}
\boxed{1}

Label 1 means the item is supported by behavioral preference. Label 0 means the evidence is insufficient or does not support behavioral relevance.

Adjudication prompt is used only when the forward and reverse annotations disagree. A neutral judge compares the two reasoning traces and decides whether the final judgment is consistent with the forward annotation. When the answer is yes, the forward label is retained for training; otherwise, the sample is discarded.

Adjudication Prompt

ROLE

You are a neutral judge for user behavior analysis in e-commerce search. Your task is to adjudicate conflicting annotations for the same query-item-user context.

TASK DEFINITION

The task is to judge whether the candidate item is supported by the user's behavioral preference under the current query intent. You should compare the forward annotation and the reverse annotation, evaluate which reasoning is better grounded in the query, item attributes, and multi-timescale behavioral context, and decide whether the forward label should be accepted.

QUERY

{query}

ITEM

Title: {item_title}

Other attributes: {item_attributes}

MULTI-TIMESCALE BEHAVIORAL CONTEXT

Real-time search interactions:
{real_time_context}

Short-term search interactions:
{short_term_context}

Long-term purchase preference:
{long_term_context}

FORWARD ANNOTATION

Forward reasoning:
{forward_reasoning}

Forward label:
{forward_label}

REVERSE ANNOTATION

Reverse reasoning:
{reverse_reasoning}

Reverse label:
{reverse_label}

OUTPUT CONSTRAINTS

Produce a concise adjudication rationale that explains which annotation is better supported by the evidence.

Then output the adjudication result in the following format:
accept_forward: {yes_or_no}

If the forward reasoning is better grounded and its label is acceptable, set accept_forward to yes. Otherwise, set accept_forward to no.

Rubric Generation Prompt

ROLE

You are an expert rubric designer for user behavior analysis in e-commerce search. Your task is to write sample-specific rubric checks for judging whether a model's reasoning is well grounded.

TASK DEFINITION

Given a query, a candidate item, the user's multi-timescale behavioral context, and the final label, generate rubric checks that evaluate whether a reasoning trace would use the provided evidence correctly. The rubrics should be sample-specific and input-aware, but they must not depend on any particular model-generated reasoning trace.

QUERY

{query}

ITEM

Title: {item_title}

Other attributes: {item_attributes}

MULTI-TIMESCALE BEHAVIORAL CONTEXT

Real-time search interactions:
{real_time_context}

Short-term search interactions:
{short_term_context}

Long-term purchase preference:
{long_term_context}

LABEL

Final label:
{label}

OUTPUT CONSTRAINTS

Generate seven rubric checks aligned with the structured reasoning steps:

1. Query intent analysis
2. Item attribute analysis
3. Real-time context analysis
4. Short-term context analysis
5. Long-term context analysis
6. Attribute-level alignment
7. Final decision grounding

Each rubric check should be specific to this sample, verifiable from the input and label, and independent of surface formatting. The weight indicates the relative importance of the question when computing the weighted rubric pass rate. Output the rubrics as a JSON list:
[
 {"step": 1, "question": "...", "weight": 1.0},
 ...
]

Rubric generation prompt is applied after a retained label is obtained. It converts the query, item, behavioral context, and label into input-aware checks aligned with the structured reasoning steps. These rubrics are used both to filter supervised reasoning traces and to compute the rubric-based reward during GRPO.

Rubric-based scoring prompt is used by the rubric judge during GRPO and reasoning-quality evaluation. Given a model output and the input-aware rubrics, the judge assigns a binary pass or fail score to each rubric item.

Rubric-Based Scoring Prompt

ROLE

You are a strict evaluator for user behavior analysis in e-commerce search. Your task is to judge whether a model-generated reasoning trace satisfies each rubric question.

INPUT

Query:
{query}

Item:

Title: {item_title}

Other attributes: {item_attributes}

Multi-timescale behavioral context:

{behavioral_context}

Model output:

{model_output}

Rubric questions:

{rubric_questions}

OUTPUT CONSTRAINTS

For each rubric question, output 1 if the model output clearly satisfies the question, and 0 otherwise. Judge each question independently. Do not give partial credit. Do not add explanations.

Output exactly the following JSON format:

```
[  
  {"question": "...", "weight": 1.0, "score": x},  
  ...  
]
```

Each score is binary: 1 for pass and 0 for fail.