

GRAM-V2: Generative Reasoning and Ranking Alignment for Large-Scale E-commerce Search

Chunyu Yuan, Ming Pang*, Yiming Deng, Xiaoyu He, Zheng Fang
Donghao Xie, Fanyi Qu, Changping Peng, Ching Law

JD.COM, Beijing, China

chunyuany93@outlook.com

{pangming8, dengyiming5, hexiaoyu5, fangzheng21, xiedonghao1, qufanyi1, pengchangping, lawching}@jd.com

Abstract

Click-through prediction is a critical task for e-commerce advertising systems. However, industrial ranking models still rely on sparse IDs and historical behavior signals, which limits their generalization to cold-start products, long-tail queries, and complex user intents. Although large language models (LLMs) provide strong semantic understanding and can generate behavioral rationales, directly using them for online ranking is impractical under strict latency constraints.

We propose GRAM-V2, a rationale-supervised ranking framework that transfers LLM-generated behavioral rationales into an efficient CTR model. GRAM-V2 uses a Teacher LLM to generate click-oriented rationales, distills them into a lightweight Student LLM, and aligns a two-tower CTR model with the Student’s representations and rationale signals. During online serving, GRAM-V2 uses only distilled representations and cached item embeddings, without generating rationales autoregressively. Large-scale offline experiments and online A/B testing show that GRAM-V2 improves CTR prediction accuracy, CTR, and advertising revenue while satisfying production latency constraints.

1 Introduction

CTR prediction is a critical component of e-commerce search advertising. Given a user, a query, and a candidate product, the ranking system must accurately estimate user engagement under strict latency constraints. Existing CTR models primarily rely on sparse IDs, statistical features, and historical behaviors (Rendle, 2010; Guo et al., 2017; Lian et al., 2018; Zhou et al., 2018). Although effective in large-scale systems, their dependence on ID-based interaction patterns limits their ability to capture fine-grained semantics, especially for cold-

start products, long-tail queries, and complex user intents.

LLMs provide strong semantic understanding and have recently been explored as feature extractors, semantic encoders, and generative recommenders for recommendation and ranking (Li et al., 2023; Lian et al., 2025; Rajput et al., 2023). However, directly applying LLMs to industrial CTR prediction remains challenging. First, generic LLM representations are not explicitly optimized for click prediction and may be misaligned with ranking objectives. Second, autoregressive reasoning introduces substantial computational overhead and is impractical for high-throughput online serving. In addition, generated rationales may appear semantically plausible without necessarily providing useful supervision for CTR prediction.

To address these challenges, GRAM-V2 extends GRAM’s query–item code alignment (Pang et al., 2025) from generative retrieval to generative ranking, distilling LLM-generated rationales into an efficient ranker. Specifically, GRAM-V2 follows a Teacher–Student–Ranker pipeline: a Teacher LLM generates click-oriented rationales from user, query, and item contexts, a lightweight Student LLM distills these rationales, and a two-tower CTR model learns from the Student’s semantic and rationale signals. During online serving, GRAM-V2 does not generate rationales autoregressively but uses the trained two-tower encoder and cached item embeddings for efficient ranking.

We evaluate GRAM-V2 on large-scale e-commerce advertising logs and real-world online traffic. GRAM-V2 consistently outperforms both traditional and LLM-enhanced CTR baselines, while delivering improvements in online CTR and advertising revenue within the production latency budget. Extensive analyses of rationale quality, model ablations, grouped AUC, and serving efficiency further demonstrate the effectiveness and deployment practicality of the proposed framework.

* Corresponding author.

Our contributions are summarized as follows:

- We propose GRAM-V2, a rationale-supervised CTR prediction framework that leverages LLM-generated behavioral rationales as training-time supervision for industrial ranking.
- We design a Teacher–Student–Ranker framework that transfers semantic and reasoning signals into an efficient two-tower CTR model without introducing autoregressive generation during online serving.
- We conduct large-scale offline and online evaluations, together with comprehensive analyses of rationale quality, alignment objectives, grouped ranking performance, and serving efficiency, demonstrating the effectiveness and practicality of GRAM-V2.

2 Related Work

2.1 CTR Prediction Models

CTR prediction has been widely studied in industrial ranking systems. Early methods, such as Factorization Machines (Rendle, 2010; Juan et al., 2017), model feature interactions over sparse categorical features, including queries, users, and item categories (Yuan et al., 2023, 2024). Deep CTR models further improve interaction modeling through neural architectures. DeepFM (Guo et al., 2017) and xDeepFM (Lian et al., 2018) capture explicit and implicit high-order feature interactions, while DIN (Zhou et al., 2018) and DIEN (Zhou et al., 2019) model dynamic user interests from historical behaviors with attention mechanisms. These methods are effective and efficient for online serving, but they still rely heavily on sparse IDs and historical interaction patterns, which limits their ability to capture fine-grained semantic relations between user intent and item attributes.

2.2 LLMs for Recommendation and Ranking

Recent studies have introduced LLMs into recommendation and ranking tasks. One line of work uses LLMs as semantic encoders or feature generators to enrich conventional recommendation models with text-based representations (Li et al., 2023; Liu et al., 2025a; Lian et al., 2025). Another line formulates recommendation and retrieval as generative tasks, where models directly generate item identifiers, retrieval codes, or bidwords (Rajput

et al., 2023; Deng et al., 2025; Chen et al., 2025; Pang et al., 2025; Liu et al., 2025b). Field Matters (Cui et al., 2026) further explores a lightweight LLM-enhanced CTR framework with field-level semantic modeling. Although these approaches improve semantic understanding, they typically use LLM outputs as auxiliary features or introduce generative components that can increase serving complexity and latency.

Different from these works, GRAM-V2 transfers reasoning and semantic knowledge from a GRPO-optimized Teacher and a compact Student to a two-tower ranker during training. The Teacher provides click-oriented rationale supervision, while the Student bridges rich reasoning signals and the lightweight ranking model. During serving, the Student supervision path, rationale decoder, and alignment modules are removed, avoiding autoregressive generation while preserving the efficiency of the original ranking architecture.

3 Model

3.1 Overview

As shown in Figure 1, GRAM-V2 consists of two phases. In Phase I, an offline Qwen3-32B Teacher is optimized via reinforcement learning to generate task-relevant behavioral rationales, which are subsequently distilled into a Qwen2.5-0.5B Student for high-throughput supervision. In Phase II, the frozen Student provides semantic and rationale signals to train a separate two-tower CTR model. During online serving, GRAM-V2 uses only the trained two-tower encoder, cached item embeddings, and interaction heads, without autoregressive rationale generation.

3.2 Prompt Definition

3.2.1 One-Tower User Prompt

For rationale generation, we construct a one-tower prompt with the user profile, recent queries, clicked products, current query, and candidate product. This complete user-query-item prompt is denoted as $Pmpt_{uqs}$.

3.2.2 Two-Tower User Prompt

For online CTR serving, the user-query side and item side are encoded separately. The user-query prompt contains the user profile, historical behaviors, and current query, but excludes the candidate product. We denote it as $Pmpt_{uq}$.

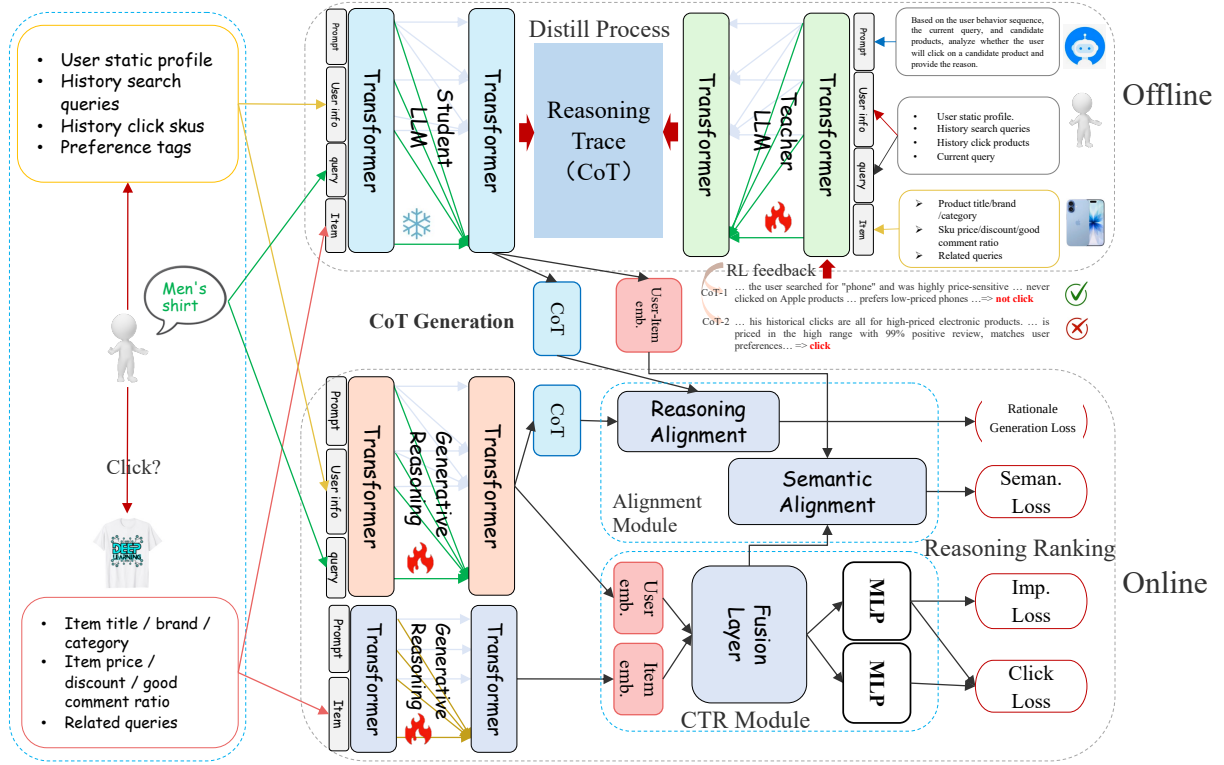


Figure 1: The architecture of GRAM-V2 for rationale-supervised CTR prediction. The rationale generation and semantic alignment losses are used only during training.

3.2.3 Item Prompt

The item prompt contains product-side information, such as title, category, brand, price, discount, rating, and other attributes. We denote it as $Pmpt_s$. Since product information changes relatively slowly, item representations are pre-computed offline and cached for online serving.

3.3 Phase I: Generative Reasoning Training

Phase I aims to obtain task-relevant rationale supervision. We first optimize a large Teacher LLM to generate rationales consistent with observed user feedback and then distill its rationale generation ability into a lightweight Student LLM for large-scale CTR training.

3.3.1 Teacher LLM Training

The Teacher is an open-source Qwen3-32B model used only offline. Given the one-tower prompt, it generates a rationale and a click prediction. For the same prompt, the model samples multiple candidate responses:

$$a_i = (c_i, \hat{y}_i), \quad i = 1, 2, \dots, G, \quad (1)$$

where c_i denotes the generated rationale, $\hat{y}_i$ denotes the predicted click label, and G is the number of sampled responses.

The response is sampled from the Teacher policy $a_i \sim \pi_\theta(\cdot | Pmpt_{uqs})$. To guide the Teacher toward useful rationale generation, we define a reward that considers both prediction correctness and response format:

$$R_i = r_{\text{reason}}(a_i) + \mathbb{I}(\hat{y}_i = y). \quad (2)$$

Here, y is the ground-truth click label, and $\mathbb{I}(\cdot)$ is the indicator function. The format reward $r_{\text{reason}}(a_i)$ is $+0.1$ when the response contains a valid analysis field and a parseable prediction, and -0.1 otherwise. The correctness term is 1 for a correct prediction and 0 otherwise. This reward encourages prediction-consistent and well-formed rationales.

Following GRPO (Shao et al., 2024), we normalize the rewards within each response group to obtain the advantage:

$$\hat{A}_i = \frac{R_i - \text{mean}(R_1, \dots, R_G)}{\text{std}(R_1, \dots, R_G) + \epsilon_{\text{norm}}}. \quad (3)$$

Our implementation follows the token-level GRPO formulation. For token $a_{i,t}$ in response a_i , the policy ratio is

$$\rho_{i,t}(\theta) = \frac{\pi_\theta(a_{i,t} | Pmpt_{uqs}, a_{i,<t})}{\pi_{\text{old}}(a_{i,t} | Pmpt_{uqs}, a_{i,<t})}. \quad (4)$$

The clipped surrogate objective averages the token-level terms in each response:

$$\begin{aligned}\bar{\rho}_{i,t}(\theta) &= \text{clip}(\rho_{i,t}(\theta), 1 - \epsilon, 1 + \epsilon), \\ \ell_i(\theta) &= \frac{1}{T_i} \sum_{t=1}^{T_i} \min \left(\rho_{i,t}(\theta) \hat{A}_i, \bar{\rho}_{i,t}(\theta) \hat{A}_i \right).\end{aligned}\quad (5)$$

The Teacher LLM is trained by minimizing the following GRPO loss:

$$\mathcal{L}_{\text{GRPO}}(\theta) = -\frac{1}{G} \sum_{i=1}^G \ell_i(\theta) + \beta_{\text{KL}} D_{\text{KL}}(\pi_{\theta} \| \pi_{\text{ref}}). \quad (6)$$

We use $G = 8$, $\epsilon = 0.2$, $\beta_{\text{KL}} = 0.01$, and $\epsilon_{\text{norm}} = 10^{-8}$. The KL term constrains the optimized policy to stay close to the reference model, which helps stabilize training and reduce degenerate rationale generation. We do not use class-balanced sampling for the GRPO stage; Section 4.4 reports diagnostics for the resulting class behavior.

3.3.2 Student LLM Distillation

Although the Teacher LLM can generate stronger rationales, using it to process all CTR training samples is computationally expensive. Therefore, we distill the Teacher into a Qwen2.5-0.5B Student. The Student uses the same one-tower prompt and learns to reproduce the Teacher-generated rationale.

Let the teacher-generated rationale be $c = [c_1, c_2, \dots, c_{L_c}]$. The Student is optimized on the distilled dataset $\mathcal{D}_{\text{SFT}}$ using the standard negative log-likelihood objective:

$$\mathcal{L}_{\text{SFT}} = -\sum_{i=1}^{L_c} \log P_{\theta}(c_i | Pmpt_{uqs}, c_{<i}). \quad (7)$$

After distillation, the Student LLM is frozen and used to provide supervision for the CTR model, including the hidden representation of the one-tower prompt and the generated rationale sequence.

3.4 Phase II: Reasoning and Ranking Alignment

Phase II transfers the Student’s semantic and rationale signals into a deployable CTR model. The motivation is that the Student observes complete user-query-item context, while the online CTR model must encode user-query and item information separately. Therefore, we align the two-tower CTR representation with the Student’s one-tower representation and the rationale generation signal.

3.4.1 CTR Prediction Module

The CTR model encodes the user-query prompt and item prompt separately:

$$\begin{aligned}\mathbf{U} &= \mathbf{LLM}_{ctr}(Pmpt_{uq}) \in \mathbb{R}^{L_u \times d}, \\ \mathbf{S} &= \mathbf{LLM}_{ctr}(Pmpt_s) \in \mathbb{R}^{L_s \times d},\end{aligned}\quad (8)$$

where L_u and L_s denote the lengths of $Pmpt_{uq}$ and $Pmpt_s$, respectively, and d is the embedding dimension. $\mathbf{LLM}_{ctr}$ is a separate Qwen2.5-0.5B copy initialized from the Student checkpoint and fine-tuned in Phase II. The user-query and item towers share encoder parameters; item representations are precomputed and cached.

We apply mean pooling to obtain fixed-length vectors:

$$\begin{aligned}\mathbf{u} &= \text{mean}(\mathbf{U}), \quad \mathbf{s} = \text{mean}(\mathbf{S}), \\ \mathbf{F} &= [\mathbf{u}; \mathbf{s}; |\mathbf{u} - \mathbf{s}|; \mathbf{u} \odot \mathbf{s}]\end{aligned}\quad (9)$$

where $\odot$ is the Hadamard product. $\mathbf{F} \in \mathbb{R}^{4d}$ encodes the similarity, relevance, and affinity between the intent of the user and the characteristics of the item.

We then employ multi-task MLP heads to jointly predict impression and click probabilities, denoted as $f_{imp}(\mathbf{F})$ and $f_{ctr}(\mathbf{F})$ respectively. The framework is optimized via joint cross-entropy losses:

$$\begin{aligned}\mathcal{L}_{imp} &= \text{CE}(y_{imp}, f_{imp}(\mathbf{F})), \\ \mathcal{L}_{ctr} &= y_{imp} \cdot \text{CE}(y_{ctr}, f_{ctr}(\mathbf{F})),\end{aligned}\quad (10)$$

where $\text{CE}(y, p) = -[y \log p + (1 - y) \log(1 - p)]$, and $y_{imp}, y_{ctr} \in \{0, 1\}$ denote the impression and click labels, respectively. The impression loss is averaged over all candidates, while the click loss is averaged over exposed samples.

3.4.2 Semantic and Reasoning Alignment

Semantic alignment transfers the Student’s complete-context representation into the two-tower CTR model. The Student LLM observes $Pmpt_{uqs}$ and captures the joint semantics of user profile, query, and candidate product, while the CTR model encodes user-query and item prompts separately for online serving. To reduce this gap, we align the item-aware two-tower interaction representation with the Student representation:

$$\begin{aligned}\mathbf{H} &= \mathbf{LLM}_{stu}(Pmpt_{uqs}), \\ \mathbf{h} &= \text{mean}(\mathbf{H}), \\ \mathcal{L}_{\text{sem}} &= \min_{\mathbf{W}} \|\mathbf{W}^T[\mathbf{u}; \mathbf{s}] - \text{sg}(\mathbf{h})\|_2^2\end{aligned}\quad (11)$$

where $\mathbf{W} \in \mathbb{R}^{2d \times d}$ and $\text{sg}(\cdot)$ denotes stop-gradient. The Student is frozen, so $\mathcal{L}_{\text{sem}}$ updates the CTR encoder and projection layer $\mathbf{W}$ but not the Student. This objective transfers complete-context semantic information while keeping two-tower online inference efficient.

In addition to semantic alignment, we use rationale generation as an auxiliary training signal. Rather than reconstructing rationales from the user-query tower alone, we condition a training-only decoder on the two-tower representation. Specifically, the decoder takes the user-query representation and the target item representation as auxiliary context: $\mathbf{x} = [\mathbf{u}; \mathbf{s}]$. The rationale generation objective is defined as:

$$P_{\theta}(c_i | \mathbf{x}, c_{<i}) = \text{Decode}_{\theta}(c_i | \mathbf{x}, c_{<i}),$$

$$\mathcal{L}_{\text{reason}} = - \sum_{i=1}^{L_c} \log P_{\theta}(c_i | \mathbf{x}, c_{<i}), \quad (12)$$

where c is the Student-generated rationale. The decoder is used only during offline training and is removed during online inference. Therefore, reasoning alignment introduces no autoregressive decoding latency during live ranking.

3.4.3 Overall Objective

Finally, the total training objective is defined as a multi-task loss function:

$$\mathcal{L} = \mathcal{L}_{\text{imp}} + \mathcal{L}_{\text{ctr}} + \alpha \mathcal{L}_{\text{sem}} + \beta \mathcal{L}_{\text{reason}}, \quad (13)$$

where α and β are hyperparameters that balance the contribution of semantic and reasoning alignment, respectively. This joint objective transfers semantic and rationale supervision from the Student LLM into the CTR model while maintaining low-latency inference for online ranking.

4 Experiments

4.1 Dataset

We evaluate GRAM-V2 on two large-scale datasets collected from a real-world e-commerce advertising platform (see Table 1). The **SFT dataset** is used for Teacher warm-up and Student distillation. The **CTR dataset** consists of seven days of impression logs, augmented with 20% randomly sampled non-exposed candidates exclusively as negative samples for the impression task to alleviate exposure bias. For CTR training, rationales are generated offline by the frozen Student LLM and used as static training-time supervision.

Table 1: Dataset statistics.

Statistics	SFT Data	CTR Data
# User-Query Pairs	3.5M	257M
# User-Product Pairs	1.28M	361M
# Uniq. Users	0.83M	35M
# Uniq. Products	0.84M	142M
Avg. # chars of query	8.0	7.0
Avg. # chars of rationale	503	510

Table 1 counts distinct pairs. The SFT data are constructed from purchase logs with clicked and non-clicked candidate sampling. One product can correspond to multiple queries, which results in more user-query pairs than user-product pairs. The CTR data come directly from impression and click logs, where one user-query request can expose multiple products; hence, the CTR data contain more user-product pairs than user-query pairs.

To prevent information leakage, dynamic features (e.g., historical CTR, conversion rate, and price) are computed strictly before the prediction timestamp. We split the CTR dataset chronologically, using the first six days for training/validation and the final day for testing.

4.2 Baseline Models

We compare GRAM-V2 with representative CTR prediction baselines, including traditional and deep CTR models, i.e., DCN (Wang et al., 2017), DeepFM (Guo et al., 2017), DIN (Zhou et al., 2018), and BERT4CTR (Wang et al., 2023), as well as recent LLM-enhanced CTR models, i.e., EGRec (Lian et al., 2025), SCTR (Chang et al., 2025), and MSD (Zhang et al., 2025). All models access the same raw feature sources, including the available textual attributes, but their representation mechanisms differ by architecture. We therefore treat comparisons with external baselines as end-to-end system comparisons. The w/o RA & SA variant retains the same raw inputs, two-tower backbone, interaction module, and optimization setting as GRAM-V2, and provides the controlled comparison for the two alignment objectives.

4.3 Evaluation Metrics

Following prior CTR prediction studies (Lian et al., 2025; Yu et al., 2025), we use AUC and LogLoss as the main metrics to evaluate ranking ability and probability calibration. We also report MAE and RMSE to measure prediction error. To assess rank-

ing within serving-relevant groups, we additionally report User-GAUC and Query-GAUC. We include only groups containing both click labels and weight each group AUC by its impression count.

4.4 Experiment Settings

All models are implemented in PyTorch. The Teacher is Qwen3-32B, the frozen Student is Qwen2.5-0.5B, and **LLM_{ctr}** is a separate Qwen2.5-0.5B copy initialized from the Student checkpoint. We train the CTR network for up to 3 epochs using AdamW with a batch size of 128 and a dropout of 0.05. The base learning rate is 5×10^{-5} for LLM layers and 1×10^{-3} for MLP heads, optimized via a linear warmup (first 10% steps) and cosine annealing. Maximum query and rationale lengths are capped at 16 and 512 tokens. Alignment weights are set to $\alpha = \beta = 1$ based on validation performance. Offline LLM inference is accelerated via TensorRT, and the best checkpoints are selected based on validation performance.

For the GRPO validation split, the ground-truth and predicted click rates are 6.2% and 8.5%, with positive- and negative-class recalls of 71.0% and 95.6%. These diagnostics indicate that the policy does not collapse to an always-negative predictor, although class-imbalance bias remains possible.

4.5 Offline Evaluation

4.5.1 Offline Performance

Table 2 compares GRAM-V2 with representative CTR baselines. Traditional ID-based models (DCN, DeepFM, DIN) mainly rely on sparse feature interactions, while BERT4CTR introduces textual representations. Recent LLM-enhanced methods such as EGRec, SCTR, and MSD further incorporate contextual semantics, but do not explicitly align reasoning signals with CTR prediction. GRAM-V2 achieves the best performance across all metrics, improving AUC from 0.8341 to 0.8436 over the strongest baseline, EGRec. To isolate the contribution of the proposed alignment objectives, we further compare against the matched w/o RA & SA variant, which uses the same two-tower backbone and inputs. Under this controlled setting, jointly applying reasoning and semantic alignment improves AUC from 0.8361 to 0.8436.

The ablation results further validate each component. Removing Reasoning Alignment or Semantic Alignment decreases AUC to 0.8401 and 0.8385, respectively, indicating that the two objectives provide complementary supervision. Replacing the

Table 2: Offline performance comparison with CTR baselines and ablation variants.

Models	AUC	LogLoss	MAE	RMSE
DCN	0.8142	0.7084	0.5508	0.5514
DeepFM	0.8185	0.7016	0.5487	0.5497
BERT4CTR	0.8306	0.6715	0.5456	0.5438
DIN	0.8317	0.6698	0.5491	0.5467
EGRec	0.8341	0.6764	0.5374	0.5341
SCTR	0.8281	0.6851	0.5418	0.5393
MSD	0.8224	0.6916	0.5382	0.5354
GRAM-V2	0.8436	0.6529	0.5341	0.5315
w/o GRPO	0.8412	0.6579	0.5362	0.5330
w/o RA	0.8401	0.6582	0.5369	0.5325
w/o SA	0.8385	0.6615	0.5392	0.5346
w/o RA & SA	0.8361	0.6627	0.5415	0.5393

GRPO-trained Teacher with an SFT Teacher also reduces AUC to 0.8412, showing the benefit of reward-optimized rationale supervision. Since all results are based on single runs, the relatively small differences among ablation variants should be interpreted as directional rather than statistically significant.

Table 3: Grouped AUC on the chronological test set. Group AUCs are weighted by impression count.

Model	User-GAUC	Query-GAUC
EGRec	0.8043	0.7951
w/o RA & SA	0.8057	0.7982
GRAM-V2	0.8132	0.8051

As shown in Table 3, GRAM-V2 improves User-GAUC and Query-GAUC over the matched w/o RA & SA model by 0.0075 and 0.0069, respectively. The improvements therefore also hold within users and queries rather than arising only from separation between easy and difficult groups.

4.6 Rationale Quality Analysis

We conduct a human evaluation on 500 randomly sampled test instances. Three domain annotators, blind to system identity, rate each generated rationale on a 1–5 ordinal scale for relevance, factuality, and consistency. Student SFT denotes the student distilled without reward-driven refinement, while GRAM-V2 denotes the student distilled from the GRPO-optimized Teacher. Ordinal Krippendorff’s α is 0.72, 0.67, and 0.74 for relevance, factuality, and consistency, respectively.

Table 4: Human evaluation of generated rationales. Scores are averaged over three annotators on a 1–5 scale.

Method	Relevance	Factuality	Consistency
Student SFT	4.12	4.05	4.08
GRAM-V2	4.43	4.31	4.36

As shown in Table 4, GRAM-V2 achieves higher scores across all dimensions, showing that reward-driven refinement improves rationale quality. We also observe failure cases in sparse-history, impulsive-click, and ambiguous-query scenarios. For example, for a broad query such as “gift”, the rationale may focus on a single recipient or occasion while overlooking other plausible intents. Therefore, rationales are used as auxiliary training-time supervision rather than hard decision rules.

4.7 Online Evaluation

4.7.1 Online Deployment

In online serving, GRAM-V2 removes the Student supervision path, rationale decoder, and alignment modules used during training. Item embeddings are precomputed and retrieved from cache, while only the user-query representation and interaction scoring are computed online. Therefore, rationale and semantic alignment are used only as offline training signals, without introducing autoregressive generation latency.

4.7.2 Online Performance

We conduct a nine-day A/B test on an online e-commerce advertising platform. GRAM-V2 serves 5% of live traffic, and the control group of the same size uses the previous production model. Traffic is randomly split at the user level, with the same retrieval, auction, and budget settings. Significance for the full-model treatment is evaluated with a two-sided user-level bootstrap test. We report relative improvements in Impressions (Imp.), CTR, CPC, and Ad Revenue.

Table 5: Relative online improvements over the production model.

Models	Imp.	CTR	CPC	Ad Rev.
Online	-	-	-	-
GRAM-V2	+0.35%	+0.56%	+0.19%	+1.10%
w/o RA	+0.11%	+0.36%	+0.04%	+0.52%
w/o SA	+0.24%	+0.20%	+0.15%	+0.58%

As shown in Table 5, GRAM-V2 improves CTR

by 0.56% and advertising revenue by 1.10% over the production model; these two full-model gains satisfy $p < 0.05$ under the user-level bootstrap test. For online ablation, each variant removes the corresponding offline alignment objective while using the same serving architecture. The directional drops of w/o RA and w/o SA are consistent with the offline ablations. The revenue gain is not solely CTR-driven. Impressions, CTR, and CPC increase by 0.35%, 0.56%, and 0.19%, respectively, whose multiplicative combination is approximately the reported 1.10% revenue gain.

4.7.3 Latency and Update Cost

Table 6 reports the deployment cost of GRAM-V2. In online serving, item embeddings are retrieved from cache, and only user-query encoding and lightweight interaction scoring are computed in real time. GRAM-V2 sustains a throughput of more than 7,600 QPS with a P99 latency of 43.2 ms, within the 50 ms production latency budget. Expensive LLM-related modules are updated offline and less frequently, while CTR alignment and item embeddings are updated daily to capture recent user behavior and product changes.

Table 6: Deployment efficiency and update cost of GRAM-V2.

Component	Use	Latency	Update
Online Model	Online	25.8 ms	-
GRAM-V2	Online	43.2 ms	-
Teacher LLM	Offline	-	Bi-weekly / 3 days
Student LLM	Offline	-	Weekly / 38 hours
CTR Training	Offline	-	Daily / 15 hours
Item Emb.	Cached	-	Daily / 5 hours

5 Conclusion and Future Work

We have presented GRAM-V2, a rationale-supervised CTR prediction framework for large-scale e-commerce search advertising. GRAM-V2 distills click-oriented rationales from an offline Teacher into a lightweight Student and transfers semantic and rationale signals into an efficient two-tower CTR model. Controlled offline comparisons, grouped AUC, and online A/B testing show improvements in CTR prediction and production metrics under the serving latency budget.

Future work will explore multimodal product understanding, more adaptive update strategies, and the accuracy-cost frontier across Student model sizes.

Limitations

Despite its effectiveness, GRAM-V2 has three main limitations. First, while distilling knowledge into a lightweight student model ensures low on-line latency, the offline training cost of the GRPO-optimized Teacher remains high. Second, in extreme cold-start or sparse-history scenarios, the generated rationales occasionally degenerate and over-explain noisy behaviors. Finally, unlike deterministic math tasks, defining a “correct” shopping rationale is inherently subjective, making automated alignment evaluation challenging.

Ethical Considerations

We follow the ACL Code of Ethics. All evaluation datasets consist of anonymized behavior logs with all Personally Identifiable Information (PII) removed to protect user privacy. To reduce potential biases inherited from pretrained LLMs, GRAM-V2 utilizes the generated rationales strictly as soft training-time auxiliary supervision rather than deterministic online routing rules, preventing stereotype amplification. Additionally, all domain annotators were fairly compensated according to local standards.

References

- Xiyu Chang, Liang Zhang, Yong He, Binzong Geng, Ke Ding, Xing Zhu, Jian Cao, and Linjian Mo. 2025. Singleton ctr prediction: Towards single-user modeling paradigm with large language models. In *Companion Proceedings of the ACM on Web Conference 2025*, pages 898–902.
- Ben Chen, Xian Guo, Siyuan Wang, Zihan Liang, Yue Lv, Yufei Ma, Xinlong Xiao, Bowen Xue, Xuxin Zhang, Ying Yang, and 1 others. 2025. Onesearch: A preliminary exploration of the unified end-to-end generative framework for e-commerce search. *arXiv*.
- Yu Cui, Feng Liu, Jiawei Chen, Xingyu Lou, Changwang Zhang, Jun Wang, Yuegang Sun, Xiaohu Yang, and Can Wang. 2026. [Field matters: A lightweight llm-enhanced method for ctr prediction](#). In *Proceedings of the ACM Web Conference 2026*, pages 6365–6376.
- Jiaxin Deng, Shiyao Wang, Kuo Cai, Lejian Ren, Qigen Hu, Weifeng Ding, Qiang Luo, and Guorui Zhou. 2025. Onerec: Unifying retrieve and rank with generative recommender and iterative preference alignment. *arXiv*.
- Huifeng Guo, TANG Ruiming, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. Deepfm: A factorization-machine based neural network for ctr prediction. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence Organization.
- Yuchin Juan, Damien Lefortier, and Olivier Chapelle. 2017. Field-aware factorization machines in a real-world online advertising system. In *Proceedings of the 26th International Conference on World Wide Web Companion*, pages 680–688.
- Xiangyang Li, Bo Chen, Lu Hou, and Ruiming Tang. 2023. Ctrl: Connect collaborative and language model for ctr prediction. *ACM Transactions on Recommender Systems*.
- 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 *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1754–1763.
- Zhaorui Lian, Binzong Geng, Xiyu Chang, Yu Zhang, Ke Ding, Ziyu Lyu, Guanghu Yuan, Chengming Li, Min Yang, Zhaoxin Huan, and 1 others. 2025. Egrec: Leveraging generative rich intents for enhanced recommendation with large language models. In *Companion Proceedings of the ACM on Web Conference 2025*, pages 1113–1117.
- Jiahao Liu, Xueshuo Yan, Dongsheng Li, Guangping Zhang, Hansu Gu, Peng Zhang, Tun Lu, Li Shang, and Ning Gu. 2025a. Improving llm-powered recommendations with personalized information. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2560–2565.
- Zhenhui Liu, Chunyuan Yuan, Ming Pang, Zheng Fang, Li Yuan, Xue Jiang, Changping Peng, Zhangang Lin, Zheng Luo, and Jingping Shao. 2025b. Multi-objective aligned bidword generation model for e-commerce search advertising. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 4299–4303.
- Ming Pang, Chunyuan Yuan, Xiaoyu He, Zheng Fang, Donghao Xie, Fanyi Qu, Xue Jiang, Changping Peng, Zhangang Lin, Zheng Luo, and 1 others. 2025. Generative retrieval and alignment model: A new paradigm for e-commerce retrieval. In *Companion Proceedings of the ACM on Web Conference 2025*, pages 413–421.
- Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan Hulikal Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Tran, Jonah Samost, and 1 others. 2023. Recommender systems with generative retrieval. *Advances in Neural Information Processing Systems*, 36:10299–10315.
- Steffen Rendle. 2010. Factorization machines. In *2010 IEEE International Conference on Data Mining*, pages 995–1000.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Dong Wang, Kavé Salamatian, Yunqing Xia, Weiwei Deng, and Qi Zhang. 2023. Bert4ctr: An efficient framework to combine pre-trained language model with non-textual features for ctr prediction. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 5039–5050.

Ruoxi Wang, Bin Fu, Gang Fu, and Mingliang Wang. 2017. Deep & cross network for ad click predictions. In *Proceedings of the ADKDD’17*, pages 1–7.

Xiaohan Yu, Li Zhang, and Chong Chen. 2025. [Explainable ctr prediction via llm reasoning](#). In *Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining*, pages 707–716.

Chunyu Yuan, Ming Pang, Zheng Fang, Xue Jiang, Changping Peng, and Zhongang Lin. 2024. A semi-supervised multi-channel graph convolutional network for query classification in e-commerce. In *Companion Proceedings of the ACM on Web Conference 2024*, pages 56–64.

Chunyu Yuan, Yiming Qiu, Mingming Li, Haiqing Hu, Songlin Wang, and Sulong Xu. 2023. A multi-granularity matching attention network for query intent classification in e-commerce retrieval. In *Companion Proceedings of the ACM Web Conference 2023*, pages 416–420.

Guoxiao Zhang, Yi Wei, Yadong Zhang, Huajian Feng, and Qiang Liu. 2025. Balancing efficiency and effectiveness: An llm-infused approach for optimized ctr prediction. In *Companion Proceedings of the ACM on Web Conference 2025*, pages 596–600.

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 *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 5941–5948.

Guorui Zhou, Xiaoqiang Zhu, Chenru 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 *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1059–1068.

A Reproducibility Details

A.1 Baseline Models

We compare GRAM-V2 with several representative CTR models. All baselines use the same available raw feature sources, while their encoders and

interaction mechanisms follow their respective architectures.

(1) Traditional and deep learning-based CTR models:

- DCN (Wang et al., 2017): It combines a deep network with a cross-network layer to model explicit and implicit feature interactions.
- DIN (Zhou et al., 2018): It uses attention over user behavior sequences to model target-related interests.
- DeepFM (Guo et al., 2017): It combines factorization machines with a deep network for feature interaction modeling.
- BERT4CTR (Wang et al., 2023): It integrates pretrained language representations with non-textual features for CTR prediction.

(2) LLM-based CTR models:

- EGRec (Lian et al., 2025): It uses LLMs to generate user-intent representations for recommendation.
- SCTR (Chang et al., 2025): It uses an LLM to model individual user behavior for CTR prediction.
- MSD (Zhang et al., 2025): It combines LLM-derived signals with a CTR model to balance computation and predictive accuracy.

A.2 Prompt Templates

To support reproducibility, Figures 2–4 show the prompt templates used by GRAM-V2. The one-tower user prompt ($P_{mpt_{uqs}}$) concatenates the user profile, recent queries, clicked products, current query, and candidate product, and is used for rationale generation and Student distillation.

1-Tower User Prompt

The user’s information:
The following user profile is known: xxx
The user’s 30 most recent historical queries: 1: xxx 2: xxx ...
The user’s 30 historical clicked products: 1: xxx 2: xxx ...

The current search query: xxx

The candidate product information: HUYEEI Tablet 2025 New 4K Large Screen Ultra-HD Full Screen 2-in-1 Deca-Core Full Network 5G WiFi ... Price: 2088.00; Price Advantage: 0.55; Discount: 0.90; Category: Tablet; Brand: HUYEE; Reviews: 39,093; Positive Rating: 99%.

Instruction: Based on the user behavior sequence, the current query, and candidate products, analyze whether the user will click on a candidate product and provide the reason.

Figure 2: One-tower user prompt ($P_{mpt_{uqs}}$) used for rationale generation and Student distillation.

The two-tower user prompt ($P_{mpt_{uq}}$) removes the candidate product and encodes only the user-query side for online serving. The item prompt

($Pmpt_s$) contains product-side attributes and is encoded and cached separately.

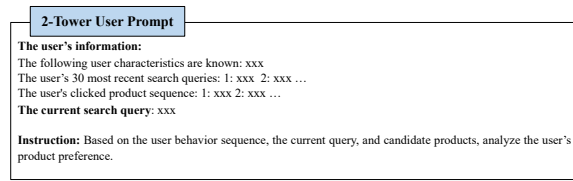


Figure 3: Two-tower user prompt ($Pmpt_{uq}$) used by the online user-query tower.

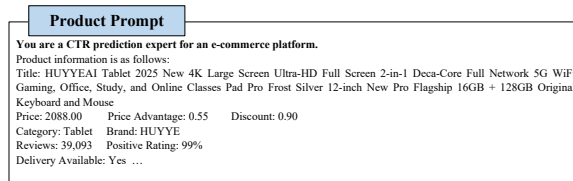


Figure 4: Item prompt ($Pmpt_s$) encoding product-side attributes.