

Towards Faithful Industrial RAG: A Reinforced Co-adaptation Framework for Advertising QA

Wenwei Li*, Ming Xu*, Tianle Xia*, Lingxiang Hu*, Yiding Sun*, Linfang Shang
Liqun Liu†, Peng Shu, Huan Yu, Jie Jiang

Tencent

{wenweiwli, flemingxu, tianlexia, lingxianghu, emanuelsun, faelynshang}@tencent.com
{liqunliu, archersh, huanyu, zeus}@tencent.com

*Equal contribution. †Corresponding author.

Abstract

Industrial advertising question answering (QA) is a high-stakes task in which hallucinated content, particularly fabricated URLs, can lead to financial loss, compliance violations, and legal risk. Although Retrieval-Augmented Generation (RAG) is widely adopted, deploying it in production remains challenging because industrial knowledge is inherently relational, frequently updated, and insufficiently aligned with generation objectives. We propose a reinforced co-adaptation framework that jointly optimizes retrieval and generation through two components: (1) Graph-aware Retrieval (GraphRAG), which models entity-relation structure over a high-citation knowledge subgraph for multi-hop, domain-specific evidence selection; and (2) evidence-constrained reinforcement learning via Group Relative Policy Optimization (GRPO) with multi-dimensional rewards covering faithfulness, style compliance, safety, and URL validity. Experiments on an internal advertising QA dataset show consistent gains across expert-judged dimensions including accuracy, completeness, and safety, while reducing the hallucination rate by 72%. A two-week online A/B test demonstrates a 28.6% increase in like rate, a 46.2% decrease in dislike rate, and a 92.7% reduction in URL hallucination. The system has been running in production for over half a year and has served millions of QA interactions.

1 Introduction

Online advertising platforms are complex, fast-evolving ecosystems where intelligent customer service (ICS) systems are critical for operational efficiency and user satisfaction (Gao et al., 2025). These systems must handle diverse intents, from pre-sales consultations to post-sales compliance appeals, under frequently updated internal policies (e.g., ad review guidelines, account systems, reimbursement protocols) that are often behind private knowledge barriers (Sharma et al., 2024).

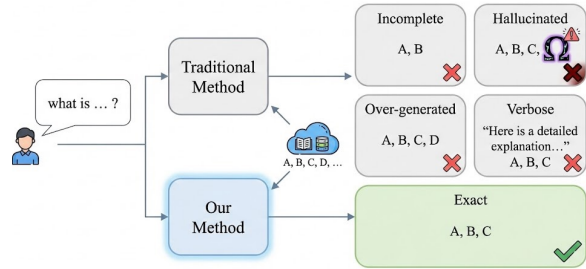


Figure 1: Traditional QA vs. our approach over a shared knowledge base. Given the same user query and knowledge items A, B, C, D, traditional methods often yield **incomplete**, **hallucinated**, **over-generated**, or **verbose** answers. Our method produces an **exact** answer that remains complete, faithful, and concise.

In this high-stakes setting, even minor factual errors can trigger compliance risks, user harm, and direct financial losses, and fabricated structured items such as URLs are particularly costly (Ji et al., 2023; Ming et al., 2025). As illustrated in Figure 1, conventional pipelines may produce incomplete, hallucinated, over-generated, or verbose answers, whereas our framework targets concise and evidence-grounded responses.

Retrieval-Augmented Generation (RAG) is a standard paradigm for grounding LLMs in external evidence (Lewis et al., 2020; Gao et al., 2024), yet production advertising question answering (QA) reveals three key gaps. First, industrial knowledge is relational and process-driven (e.g., products, rules, procedures), where single-shot hybrid retrieval can miss multi-hop dependencies; graph-based retrieval such as GraphRAG addresses this by explicitly modeling entities and relations for cross-document reasoning (Edge et al., 2024). Second, simply expanding context length is insufficient: under the “lost in the middle” effect, models may fail to reliably use evidence in long inputs, motivating targeted evidence selection (Liu et al., 2023). Third, generation must satisfy strict style and compliance

constraints, yet even strong models may deviate from provided context under unanswerable or counterfactual inputs (Ming et al., 2025; Rakin et al., 2024); while reinforcement learning (Schulman et al., 2017a; Rafailov et al., 2023) can align generation with task constraints, and post-hoc methods such as SelfCheckGPT (Manakul et al., 2023) can detect unsupported content, treating retrieval and generation as isolated stages leaves a coordination gap. Moreover, common public benchmarks emphasize static factoid QA and retrieval accuracy, whereas industrial deployment requires faithfulness under evolving knowledge, policy-sensitive cases, and strict latency budgets.

To address these gaps, we propose an end-to-end reinforced co-adaptation framework that jointly optimizes retrieval and evidence-grounded generation. It has two key components: (1) **Graph-aware Retrieval** via GraphRAG, which models relationships between products, rules, and processes to support multi-hop reasoning and terminology alignment (Edge et al., 2024); and (2) **Evidence-constrained Reinforcement Learning (RL)**, which aligns the generator with retrieved evidence using multi-dimensional rewards that encourage faithfulness while enforcing style, safety, and URL validity.

Our contributions are as follows:

- We propose a co-adaptation framework that jointly optimizes GraphRAG-based retrieval and an RL-tuned generator, achieving superior alignment between retrieved domain knowledge and generated responses.
- We design a multi-dimensional RL objective covering faithfulness, style compliance, safety, and URL validity, explicitly penalizing unsupported content and hallucinated links.
- We deploy the system on a large-scale advertising platform, serving millions of QA interactions over half a year. A two-week A/B test shows a 28.6% like-rate increase, a 46.2% dislike-rate reduction, and a 92.7% reduction in URL hallucination.

2 Methodology

2.1 Problem Formulation

We formulate advertising QA as a constrained conditional generation task (Figure 2). Given a user

query q and a dynamically updated private knowledge base K , the system retrieves a relevant evidence set $D = \{d_1, \dots, d_k\}$. The generator π_θ produces a response A that maximizes $P(A | q, D)$ subject to constraints C , including zero URL hallucinations, domain-specific style compliance, and safety requirements.

2.2 Graph-aware Retrieval

To address the limitations of traditional hybrid retrieval methods (e.g., BGE + BM25) in handling complex multi-hop dependencies and domain-specific terminology, we propose a Graph-aware Retrieval module that integrates GraphRAG with a carefully curated, high-citation knowledge base, complemented by a fused dual-branch retrieval architecture for industrial-scale deployment.

High-Citation Knowledge Base. GraphRAG enhances retrieval but introduces substantial computational overhead. To balance effectiveness and efficiency, we maintain a high-citation knowledge base $K_h \subset K$ through traffic-driven feedback. We accumulate recall frequency for each knowledge chunk from production query logs as a “citation heat” indicator, and periodically select the top- $N\%$ most frequently cited items to form K_h . This curated subset serves as the subgraph for GraphRAG, reducing traversal complexity while preserving effectiveness through automatic rolling updates. In the snapshot used for our main experiments, the resulting graph contains 4,686 entities, 10,712 relations, and 12,809 total nodes after adding community nodes, which provides enough relational coverage for frequent production queries while keeping traversal latency manageable.

GraphRAG Architecture. We construct a knowledge graph $G = (V, E)$ over K_h via entity extraction and relation identification, with community detection partitioning the graph into hierarchical subgraphs for semantic aggregation. The retrieval layer supports dynamic routing between hybrid retrieval and graph-based traversal, balancing efficiency for simple queries with multi-hop reasoning for complex ones. High-citation subgraph pruning constrains retrieval scope, and incremental updates maintain temporal currency without full reconstruction.

Parallel Retrieval Architecture. To mitigate GraphRAG latency while maximizing recall, we execute the GraphRAG branch and the traditional

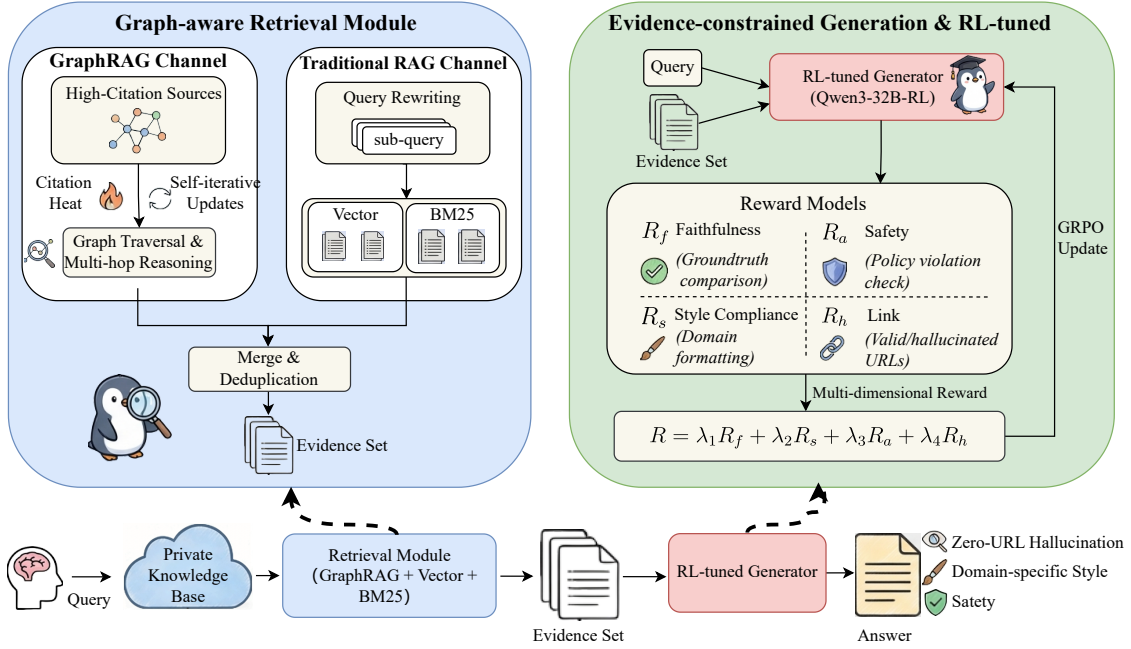


Figure 2: System overview. Given a user query q and a private knowledge base K , the retrieval system constructs an evidence set D via two parallel channels: a GraphRAG channel over a high-citation knowledge base K_h and a traditional RAG channel with query rewriting and BGE + BM25 hybrid retrieval. Results are merged and deduplicated. The RL-tuned generator then produces a response optimized by GRPO with multi-dimensional rewards for faithfulness, style compliance, safety, and URL validity.

RAG branch concurrently. The GraphRAG branch performs asynchronous traversal over K_h for multi-hop reasoning, while the traditional branch uses BGE + BM25 hybrid retrieval with multi-path query rewriting that decomposes complex queries into parallel sub-queries. We then merge the two evidence streams, deduplicate candidates using canonical document identifiers together with normalized text matching, and rerank the merged set with a lightweight Qwen3-4B reranker before truncation. We refer to this final fused retrieval system as **G-RAG+** in the experiments, to distinguish it from the standalone GraphRAG branch.

2.3 Evidence-constrained Generation

The generation module centers on an RL-tuned generator (Qwen3-32B-RL). While supervised fine-tuning establishes foundational formatting, reinforcement learning is critical for steering the model toward stable, safe, and hallucination-free responses under strict industrial constraints.

We optimize the generator using GRPO (Shao et al., 2024), whose group-based mechanism stabilizes training under noisy reward signals. Unlike PPO (Schulman et al., 2017b), which requires a separate critic model, GRPO estimates the baseline from group rewards, reducing memory over-

head and training instability. This is particularly valuable for industrial applications where reward signals are inherently noisy due to the subjective nature of style and safety assessments.

We design a multi-dimensional reward function:

$$R = \lambda_1 R_f + \lambda_2 R_s + \lambda_3 R_a + \lambda_4 R_h \quad (1)$$

where λ_i are weighting coefficients. Following preliminary experiments, we set $\lambda_3 = 2.0$ and $\lambda_4 = 2.0$ to prioritize safety and hallucination reduction, with $\lambda_1 = \lambda_2 = 1.0$. The reward components are:

- **Evidence Faithfulness (R_f):** Training-time proxy signal measuring alignment with the ground-truth answer via pairwise LLM-as-judge comparison.
- **Style Compliance (R_s):** Training-time proxy signal evaluating adherence to advertising domain conventions, including tone, professionalism, and formatting.
- **Safety (R_a):** Training-time proxy signal detecting platform policy violations and ensuring regulatory compliance.
- **URL Validity (R_h):** Rewards valid URLs and penalizes hallucinated ones. A URL is valid if it appears in the evidence D , or if its prefix belongs

to an approved pool and its HTTP status code is in $\{200, 301, 302\}$.

We selected the weights through a grid search over 15 completed settings, including balanced and safety/link-emphasized variants. All settings improved total reward over the base model by roughly 220%–240%, and the top configurations were within 5% of each other, indicating that the method is robust rather than dependent on a narrow optimum. We therefore choose $(\lambda_1, \lambda_2, \lambda_3, \lambda_4) = (1, 1, 2, 2)$ because it yields the best trade-off between factual utility, safety, and link validity. More adaptive reward-balancing strategies are also plausible extensions, as recent RL-based systems have explored dynamic weighting and task-specific reward design in related settings (Yang, 2026; Qian, 2025). Appendix A further reports reward ablations and additional matched-baseline comparisons. The full reward computation procedure, including URL extraction and validation details, is provided in Algorithm 1 in the Appendix.

3 Experiments

We evaluate our approach using both offline and online metrics.

3.1 Experimental Setting

Dataset. We evaluate on the **Advertising QA Dataset**, an internal Chinese advertising customer-service dataset with 3,000 expert-annotated question–answer pairs covering product rules, operational procedures, policy interpretation, and account-management scenarios. For out-of-domain generalization, we use **FaithEval** (Ming et al., 2025), which tests faithfulness under unanswerable questions, counterfactual contexts, and inconsistent information. We choose FaithEval because it directly probes the failure modes most relevant to faithful industrial QA, whereas many public RAG benchmarks focus more on static fact retrieval or multi-hop reasoning with relatively explicit evidence and do not reflect evolving policies, link validity, or safety-sensitive refusal behavior.

Evaluation Protocol. We compare systems along two axes: (i) **retrieval**, where we contrast Base RAG (a standard RAG pipeline with a reranker) with **G-RAG+** (our fused retrieval system combining the GraphRAG branch and the traditional branch), and (ii) **generation backbones**, where we evaluate open-source and proprietary models as well as our RL-tuned model.

We use a hybrid evaluation protocol. ROUGE-L (0–100) is computed automatically, while Accuracy, Completeness, Clarity, Style, and Safety are rated by human experts on a 0–10 scale. Hallucination Rate is also assessed by human experts at the case level: for each case, if the answer contains any fabricated or unsupported content, we count it as one hallucinated case. The LLM judge is used only during RL reward construction and auxiliary analysis, not for the main offline results in Table 1. Formally, given N cases and an indicator $\mathbb{I}[\cdot]$, we report

$$\text{HR} = \frac{1}{N} \sum_{i=1}^N \mathbb{I}[\text{answer}_i \text{ contains hallucination}].$$

ROUGE-L measures lexical overlap with the reference; lower HR indicates fewer hallucinated cases.

Models. We evaluate five representative backbones: DeepSeek-V3.2 (DeepSeek-AI et al., 2025), GPT-5.2 (OpenAI, 2025), Qwen3-32B (Qwen Team, 2025), Qwen3-32B-SFT, and Qwen3-32B-RL (ours). All evaluated models support reasoning capabilities; to match production latency constraints, we evaluate all models in non-thinking mode for a fair comparison.

This model set covers strong open-source and commercial baselines, isolates the impact of RL (vs. SFT) on the same backbone, and tests robustness across model families.

3.2 Main Results

Table 1 reports the main offline results. Replacing Base RAG with G-RAG+ consistently improves quality and reduces hallucinations. DeepSeek-V3.2 improves ROUGE-L from 33.27 to 37.00 (+3.73) and reduces Hallucination Rate from 0.0077 to 0.0030 (61% relative reduction). Similar patterns hold for GPT-5.2 (ROUGE-L: 32.82→35.88; Hallucination Rate: 0.0057→0.0023, 60%) and Qwen3-32B (ROUGE-L: 29.39→32.96; Hallucination Rate: 0.0117→0.0060). These gains indicate that graph-aware multi-hop evidence aggregation is most effective when coupled with the traditional retrieval branch rather than used in isolation.

RL provides additional gains beyond retrieval improvements. Under G-RAG+, Qwen3-32B-RL (Ours) improves ROUGE-L from 33.82 to 35.49 over Qwen3-32B-SFT (+1.67), and lowers Hallucination Rate from 0.0047 to 0.0013 (72% relative reduction). Even under Base RAG, Qwen3-32B-RL

achieves a 0.0017 Hallucination Rate, indicating that evidence-constrained RL targets hallucination behaviors that supervised fine-tuning alone cannot eliminate. The absolute gains on utility metrics are moderate, but they are accompanied by large reliability improvements, especially for hallucination control and safety, which is the primary optimization target in production.

The complementary effect between GraphRAG and RL is evident across all metrics. GraphRAG primarily improves coverage-related metrics (ROUGE-L, Completeness), while RL enhances reliability and compliance metrics (Style, Safety, Hallucination Rate). Their combination achieves the best overall performance, with our final system outperforming the strongest baseline (DeepSeek-V3.2 with G-RAG+) on Hallucination Rate (0.0030 vs. 0.0013) while maintaining competitive quality scores. Appendix A provides additional same-generator comparisons against stronger retrieval baselines together with reward ablations.

3.3 G-RAG+ Effectiveness

We evaluate the fused retrieval module both offline and online.

Offline Evaluation. We assess retrieval via side-by-side expert comparison and knowledge recall analysis.

Knowledge Recall Enhancement. Figure 3 shows progressive improvements in knowledge recall. Effective knowledge chunks per query increase from 3.9 under Base RAG to 4.5 with the standalone GraphRAG branch and to 6.3 with G-RAG+, a 61.5% overall improvement. Recall effectiveness correspondingly rises from 73.6% to 90.5%, showing that the fused retriever benefits from combining relation-aware graph exploration with direct hybrid recall.

Retrieval Quality Optimization. In expert evaluation, the Good:Same:Bad ratio reaches 32.4%:64.9%:2.7% at retrieval. The Good ratio is 12 \times higher than Bad, indicating effective noise filtering.

End-to-End Performance. The end-to-end Good:Same:Bad ratio reaches 24.3%:71.6%:4.1%, with positive gains outweighing negative impacts by 6 \times .

Online A/B Testing. We deployed at 50% traffic. Table 2 shows consistent improvements: like rate increases from 0.21% to 0.27% (+28.6%), dislike rate decreases from 0.26% to 0.18% (-30.8%), and

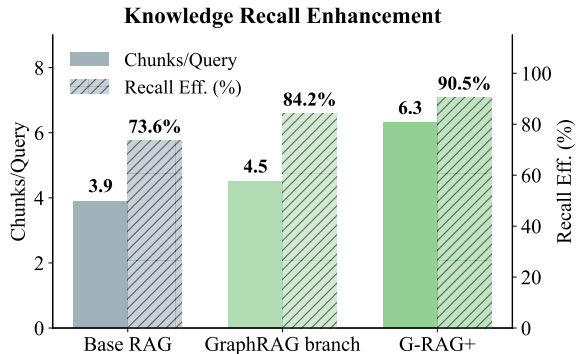


Figure 3: Knowledge recall enhancement across Base RAG, the standalone GraphRAG branch, and G-RAG+. Effective chunks per query and recall effectiveness in percent.

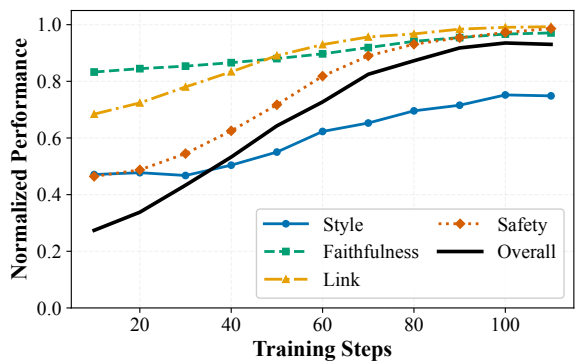


Figure 4: Training dynamics of multi-dimensional reward components during RL.

average conversation turns increase from 1.54 to 1.81 (+17.5%), indicating improved user engagement.

3.4 RL Reward Effectiveness

Figure 4 shows consistent improvement across all reward components during RL fine-tuning. With only 1,000 training samples, all metrics improve rapidly within 100 steps and then stabilize, indicating an efficient reward design. This data scale is plausible because retrieval remains responsible for most knowledge access, while RL mainly aligns response behavior with reliability requirements such as evidence usage, safety, and URL validity. This interpretation is consistent with recent post-training evidence that carefully selected RL data can outperform much larger sets (Li et al., 2025), that difficulty-calibrated RL recipes can matter as much as raw scale (An et al., 2025), and that RL often updates only sparse but effective subnetworks (Mukherjee et al., 2025). We therefore curate the 1K SFT and 1K RL samples toward hard cases

Metric	DeepSeek-V3.2		GPT-5.2		Qwen3-32B		Qwen3-32B-SFT		Qwen3-32B-RL	
	Base RAG	G-RAG+	Base RAG	G-RAG+	Base RAG	G-RAG+	Base RAG	G-RAG+	Base RAG	Ours
ROUGE-L \uparrow	33.27	37.00	32.82	<u>35.88</u>	29.39	32.96	30.79	33.82	31.40	35.49
Accuracy \uparrow	7.82	<u>8.37</u>	7.94	8.39	7.25	7.82	7.50	8.10	7.85	8.26
Completeness \uparrow	6.78	7.10	6.70	<u>7.08</u>	6.20	6.66	6.43	6.91	6.46	6.99
Clarity \uparrow	8.96	8.99	8.92	<u>8.97</u>	8.52	8.74	8.82	8.94	8.83	8.95
Style \uparrow	8.19	8.25	8.14	<u>8.25</u>	7.82	8.03	8.07	8.19	<u>8.27</u>	8.33
Safety \uparrow	9.94	9.93	9.95	9.96	9.88	9.91	9.95	9.94	<u>9.97</u>	9.99
Hallucination Rate \downarrow	0.0077	0.0030	0.0057	0.0023	0.0117	0.0060	0.0117	0.0047	<u>0.0017</u>	0.0013

Table 1: Main experimental results. **Ours** refers to Qwen3-32B-RL with G-RAG+. Best results in **bold**, second best underlined.

Metric	Base RAG	Ours	Δ
Like Rate (%)	0.21	0.27	+28.6%
Dislike Rate (%)	0.26	0.18	-30.8%
Avg. Conv. Turns	1.54	1.81	+17.5%

Table 2: Online A/B testing at 50% traffic.

that expose hallucination, refusal, and formatting failures instead of relying on broad but redundant supervision.

The reward components exhibit distinct optimization patterns. Faithfulness and URL validity rewards show the steepest initial ascent, indicating that the model quickly learns to align with retrieved evidence and avoid hallucinated links. Style and safety rewards improve more gradually, reflecting the nuanced nature of domain-specific tone and compliance requirements. The overall reward converges to a stable high value, suggesting that the multi-objective optimization achieves balanced improvements across all dimensions without detrimental trade-offs.

3.5 Generalization on FaithEval

To assess whether our RL-tuned model generalizes beyond the in-domain setting, we evaluate on **FaithEval**. Figure 5 shows the results. We view this benchmark as a complementary stress test for faithful response behavior rather than a full proxy for industrial deployment, because it captures missing, conflicting, and misleading evidence but does not include online knowledge evolution or platform-specific safety constraints.

Our RL-tuned model improves over Qwen3-32B on all FaithEval subsets: Unanswerable 44.60% \rightarrow 53.40%, Counterfactual 57.90% \rightarrow 64.40% (outperforming DeepSeek-V3.2 at 56.40%), and Inconsistent 63.80% \rightarrow 84.60%. The gains on Unanswerable and Counterfactual suggest stronger refusal behavior when context is missing or misleading. On Inconsistent, it reaches

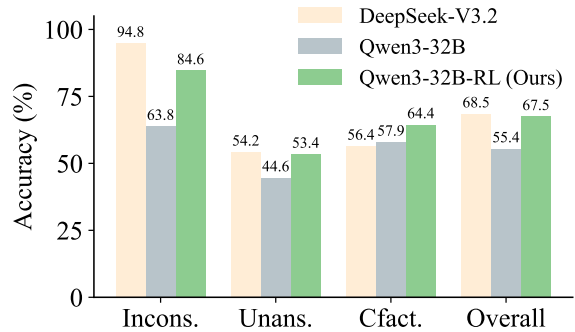


Figure 5: FaithEval generalization: accuracy (%) on Inconsistent, Unanswerable, Counterfactual, and Overall.

84.60%, substantially above Qwen3-32B (63.80%) and closer to DeepSeek-V3.2 (94.80%). These results indicate improved contextual faithfulness without degrading generalization.

Scope clarification. No single public benchmark fully matches the complete production setting studied here, which jointly involves evolving private knowledge, safety constraints, domain-specific style, and URL-validity control. We therefore use public benchmarks to test transferable subproblems, while relying on the internal Advertising QA benchmark and online A/B testing to evaluate the full industrial objective end to end.

3.6 Production Deployment

3.6.1 Offline Evaluation

We compare against a Base RAG + DeepSeek-V3 (Liu et al., 2024) baseline via expert assessment on completeness, professionalism, compliance, and hallucination. As shown in Figure 6, our method wins substantially more often than it loses, with the largest gains in professionalism (45.2% win) and compliance (41.9% win) and a low loss rate (1.1%). It also improves hallucination outcomes (7.7% win vs. 0.1% loss), supporting the benefit of co-adapting GraphRAG and RL-tuned generation.

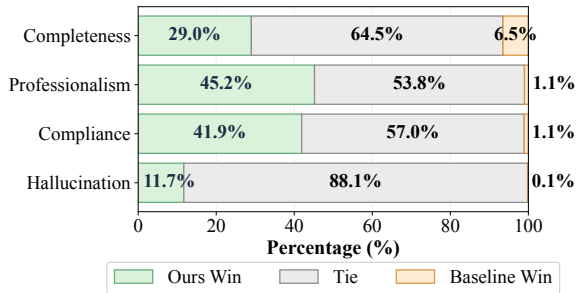


Figure 6: Offline evaluation comparison: win/tie/lose distribution across four dimensions.

Metric	Baseline	Ours	Δ
Like Rate (%)	0.21	0.27	+28.6%
Dislike Rate (%)	0.26	0.14	-46.2%
URL Hallu. (%)	0.0041	0.0003	-92.7%
Latency (s)	2.5	3.1	+24.0%

Table 3: Online A/B testing results (two weeks, 50% traffic).

3.6.2 Online A/B Testing

A two-week online A/B test compares our deployed system against the Base RAG + DeepSeek-V3 baseline, with a 50%/50% traffic split (Table 3). Our method increases like rate from 0.21% to 0.27% (+28.6%), decreases dislike rate from 0.26% to 0.14% (-46.2%), and reduces URL hallucination from 0.0041% to 0.0003% (-92.7%). Average first-token latency rises from 2.5s to 3.1s (+24.0%), which remains acceptable in practice. Overall, the A/B results suggest that reinforced co-adaptation improves both user satisfaction and reliability under real traffic, with a manageable latency trade-off.

3.6.3 Latency Analysis

Table 4 details the latency distribution. Query rewriting takes 690ms; within G-RAG+, the GraphRAG branch takes 852ms and the BGE + BM25 branch takes 167ms; reranking takes 557ms; generation takes 801ms; and safety guardrails take 230ms. Total latency is 3130ms, meeting acceptable thresholds for user experience and industrial deployment.

The latency breakdown highlights several avenues for optimization. GraphRAG retrieval incurs the largest single latency cost (852ms), which motivates our high-citation knowledge base design to constrain graph traversal and reduce overhead. Executing GraphRAG in parallel with the BGE + BM25 pipeline ensures that the slower graph-based retrieval does not block the faster hybrid channel.

Module	Latency (ms)
Query Rewriting	690
GraphRAG Retrieval	852
BGE + BM25 Retrieval	167
Reranking	557
Generation	801
Safety Guardrails	230
Total	3130

Table 4: Latency breakdown by module.

Generation latency (801ms) is on par with standard large language model inference, suggesting that RL fine-tuning does not introduce noticeable computational overhead relative to the base model. Safety guardrails incur an additional 230ms as post-processing, without affecting time-to-first-token, thereby preserving system responsiveness.

4 Conclusion

We present a reinforced co-adaptation framework to mitigate hallucinations in industrial advertising Q&A by jointly optimizing GraphRAG and an RL-tuned generator guided by multi-dimensional rewards, thereby narrowing the retrieval-generation gap and reducing unsupported content and hallucinated or invalid links. Our results show that graph-aware retrieval with a high-citation knowledge base balances multi-hop evidence aggregation with computational efficiency, while evidence-constrained RL further suppresses hallucinations without sacrificing domain style compliance or safety. Extensive offline evaluations, a two-week production A/B test, and over six months of deployment collectively validate that the approach improves answer reliability and user-facing quality at scale under practical latency constraints.

Ethics Statement

Our research targets high-stakes industrial advertising question answering and adheres to ethical principles that prioritize user rights. We aim to improve system reliability and safety by reducing unsupported claims and hallucinated or invalid URLs that could mislead users or introduce compliance risks. Any dataset examples are used solely for scientific analysis and do not necessarily reflect the views of the authors. All resources are intended for scientific research purposes only, contributing to the development of more secure and reliable digital platforms.

References

- Chenxin An, Zhihui Xie, Xiaonan Li, Lei Li, Jun Zhang, Shansan Gong, Ming Zhong, Jingjing Xu, Xipeng Qiu, Mingxuan Wang, and Lingpeng Kong. 2025. [POLARIS: A post-training recipe for scaling reinforcement learning on advanced reasoning models](#). Project blog, accessed 2026-04-20.
- CircleMind AI. 2024. [Fast-GraphRAG](#). GitHub repository, accessed 2026-04-20.
- DeepSeek-AI, Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, and 1 others. 2025. [Deepseek-v3.2: Pushing the frontier of open large language models](#). *arXiv preprint arXiv:2512.02556*.
- Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitanaky, Robert Osazuwa Ness, and Jonathan Larson. 2024. [From local to global: A graph RAG approach to query-focused summarization](#). *arXiv preprint arXiv:2404.16130*.
- Tianhong Gao, Jundong Shen, Jiapeng Wang, Bei Shi, Ying Ju, Junfeng Yao, and Huiyu Yu. 2025. [Benchmarking and learning real-world customer service dialogue](#). *arXiv preprint arXiv:2510.22143*.
- Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024. [Retrieval-augmented generation for large language models: A survey](#). *arXiv preprint arXiv:2312.10997*.
- Gemini Team, Google. 2025. [Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities](#). Technical report, accessed 2026-02-08.
- Zirui Guo, Lianghao Zhu, Lei Liu, Yihong Chen, Dongping Chang, Xuemin Yin, Yuzhong Qu, Yang Yuan, and Daxin Zhang. 2024. [LightRAG: Simple and fast retrieval-augmented generation](#). *arXiv preprint arXiv:2410.05779*.
- Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. [Survey of hallucination in natural language generation](#). *ACM Computing Surveys*, 55(12):1–38.
- Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. [Efficient memory management for large language model serving with PagedAttention](#). *arXiv preprint arXiv:2309.06180*.
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. [Retrieval-augmented generation for knowledge-intensive NLP tasks](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474.
- Xuefeng Li, Haoyang Zou, and Pengfei Liu. 2025. [LIMR: Less is more for RL scaling](#). *arXiv preprint arXiv:2502.11886*.
- Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024. [Deepseek-v3 technical report](#). *arXiv preprint arXiv:2412.19437*.
- Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. [Lost in the middle: How language models use long contexts](#). *arXiv preprint arXiv:2307.03172*. Accepted to TACL 2023.
- Potsawee Manakul, Adian Liusie, and Mark Gales. 2023. [Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9004–9017, Singapore. Association for Computational Linguistics.
- Yifei Ming, Senthil Purushwalkam, Shrey Pandit, Zixuan Ke, Xuan-Phi Nguyen, Caiming Xiong, and Shafiq Joty. 2025. [Faitheval: Can your language model stay faithful to context, even if “the moon is made of marshmallows”](#). In *International Conference on Learning Representations*.
- Sagnik Mukherjee, Lifan Yuan, Dilek Hakkani-Tur, and Hao Peng. 2025. [Reinforcement learning finetunes small subnetworks in large language models](#). *arXiv preprint arXiv:2505.11711*.
- OpenAI. 2025. [Introducing gpt-5.2](#). Accessed: 2026-02-08.
- Cheng Qian. 2025. [ToolRL: Reward is all tool learning needs](#). *arXiv preprint arXiv:2504.13958*.
- Qwen Team. 2025. [Qwen3-32b](#). Hugging Face model, accessed 2026-02-08.
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2023. [Direct preference optimization: Your language model is secretly a reward model](#). In *Advances in Neural Information Processing Systems*.
- Salman Rakin, Md. A. R. Shibly, Zahin M. Hossain, Zeeshan Khan, and Md. Mostofa Akbar. 2024. [Leveraging the domain adaptation of retrieval augmented generation models for question answering and reducing hallucination](#). *arXiv preprint arXiv:2410.17783*.
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017a. [Proximal policy optimization algorithms](#). *arXiv preprint arXiv:1707.06347*.

- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017b. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.
- Sanat Sharma, David Seunghyun Yoon, Franck Dernoncourt, Dewang Sultania, Karishma Bagga, Mengjiao Zhang, Trung Bui, and Varun Kotte. 2024. Retrieval augmented generation for domain-specific question answering. *arXiv preprint arXiv:2404.14760*.
- Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2024. HybridFlow: A flexible and efficient RLHF framework. *arXiv preprint arXiv:2409.19256*.
- Tencent Hunyuan Team. 2025. Hunyuan-TurboS: Advancing large language models through mamba-transformer synergy and adaptive chain-of-thought. *Preprint*, arXiv:2505.15431.
- Ke Yang. 2026. Deep GraphRAG: A balanced approach to hierarchical retrieval and adaptive integration. *arXiv preprint arXiv:2601.11144*.
- Qiyang Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, and 16 others. 2025. DAPO: An open-source LLM reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*.
- Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, and 1 others. 2025. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*.
- Yuze Zhao, Jintao Huang, Jinghan Hu, Xingjun Wang, Yunlin Mao, Daoze Zhang, Zeyinzi Jiang, Zhikai Wu, Baole Ai, Ang Wang, Wenmeng Zhou, and Yingda Chen. 2025. Swift: A scalable lightweight infrastructure for fine-tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 29733–29735.

A Implementation Details

This section details the parameter settings, end-to-end pipeline implementation, and training specifics that instantiate the method described in the main paper.

Query rewriting and retrieval. Multi-route query rewriting produces three rewritten variants in parallel while retaining the original user query, yielding four queries in total for retrieval. The GraphRAG component follows the standard Microsoft GraphRAG design (Edge et al., 2024), with local search used for graph traversal. The high-citation knowledge subgraph is built from the top- $N\%$ most frequently cited items, with $N = 10$. The traditional RAG channel uses hybrid retrieval with BGE + BM25 (run jointly); results from both channels are merged and deduplicated, then reranked by a lightweight Qwen3-4B reranker (Zhang et al., 2025). Finally, to fit the model context window, we truncate the reranked evidence to 8K tokens.

Graph statistic	Value
Entities	4,686
Relations / edges	10,712
Total nodes (incl. communities)	12,809

Table 5: Snapshot of the high-citation GraphRAG subgraph used in the main experiments.

SFT stage. The first stage is supervised fine-tuning with LoRA on the base model Qwen3-32B, implemented with the SWIFT infrastructure (Zhao et al., 2025). We use a learning rate of 1×10^{-4} , train for 5 epochs on $8 \times$ NVIDIA H20 GPUs, and use 1k human-annotated dialogue samples.

RL stage. The second stage uses reinforcement learning via the VERL framework (Sheng et al., 2024) and the GRPO algorithm. Training is again LoRA-based on $16 \times$ H20 GPUs, with 1k prompts and responses labeled by Gemini 2.5 Pro (Gemini Team, Google, 2025) for reward learning. The judge used to compute rewards is Hunyuan TurboS (Tencent Hunyuan Team, 2025). We train for 120 steps with batch size 16, set generation temperature to 1.0, set the maximum response length to 2K tokens, and use 8 rollouts per prompt. All reward terms are normalized before combination. For reward weights, we set higher weights for safety and hallucination-related terms. We set $\lambda_3 = 2.0$ for safety and $\lambda_4 = 2.0$ for hallucination and link

penalty, while other weights are set to 1. Across 15 completed coefficient settings, all runs improved the total reward over the base model by roughly 220%–240%, and the top settings were within 5% of each other. Following a DAPO-style setup (Yu et al., 2025), the reference-model KL term is removed.

Safety guardrails. During streaming generation, safety guardrails post-process the output to detect and filter policy violations and hallucinated URLs before serving, enforcing zero-hallucination and strict safety constraints in the final response.

B Reward Computation Algorithm

Algorithm 1 Multi-dimensional Reward Computation

Require: Generated answer A , retrieved evidence $D = \{d_1, \dots, d_k\}$, ground truth answer A_{gt} , URL prefix candidate pool \mathcal{C}_p

Ensure: Total reward R

- 1: Extract URLs via regex: $\mathcal{U} \leftarrow \text{ExtractURLs}_{re}(A)$
 - 2: Extract evidence URLs via regex: $\mathcal{U}_D \leftarrow \text{ExtractURLs}_{re}(D)$
 - 3: HTTP status set: $\mathcal{S} \leftarrow \{200, 301, 302\}$
 - 4: URLs in evidence: $\mathcal{U}_{evi} \leftarrow \mathcal{U} \cap \mathcal{U}_D$
 - 5: URLs not in evidence: $\mathcal{U}_{out} \leftarrow \mathcal{U} \setminus \mathcal{U}_D$
 - 6: Prefix-approved URLs: $\mathcal{U}_{pref} \leftarrow \{u \in \mathcal{U}_{out} \mid \text{Prefix}(u) \in \mathcal{C}_p\}$
 - 7: HTTP-valid URLs: $\mathcal{U}_{http} \leftarrow \{u \in \mathcal{U}_{pref} \mid \text{code}(u) \in \mathcal{S}\}$
 - 8: Valid URLs: $\mathcal{U}_{valid} \leftarrow \mathcal{U}_{evi} \cup \mathcal{U}_{http}$
 - 9: $R_f \leftarrow f_{\text{faithful}}(A, A_{gt})$ {Training-time faithfulness proxy via LLM judge}
 - 10: $R_s \leftarrow f_{\text{style}}(A)$ {Training-time style proxy via LLM judge}
 - 11: $R_a \leftarrow f_{\text{safety}}(A)$ {Training-time safety proxy via LLM judge}
 - 12: $R_h^+ \leftarrow \text{Reward}(\mathcal{U}_{valid})$ {Positive reward for valid links}
 - 13: $R_h^- \leftarrow \text{Penalty}(\mathcal{U} \setminus \mathcal{U}_{valid})$ {Negative penalty for invalid links}
 - 14: $R_h \leftarrow R_h^+ - R_h^-$
 - 15: $R \leftarrow \lambda_1 R_f + \lambda_2 R_s + \lambda_3 R_a + \lambda_4 R_h$
 - 16: **return** R
-

C Additional Controlled Analyses

Stronger retrieval baselines under matched settings. To supplement the main comparison

against Base RAG, we additionally compare our retrieval stack with two recent retrieval-oriented baselines, LightRAG (Guo et al., 2024) and Fast-GraphRAG (CircleMind AI, 2024), under matched corpus and evaluation settings using pairwise preference evaluation. Table 6 reports the round-robin summary. Our system achieves the best overall quality-efficiency trade-off, outperforming Fast-GraphRAG clearly and remaining competitive with LightRAG while running substantially faster.

Method	vs. Light	vs. Fast	Pts	Lat.
Ours	25:29:20	26:32:16	81.5	0.7s
LightRAG	–	29:34:11	80.5	2.4s
Fast-GraphRAG	11:34:29	–	60.0	2.7s

Table 6: Round-robin summary against stronger retrieval baselines under matched settings. Entries are reported as win:tie:loss counts in pairwise preference evaluation.

Same-backbone attribution. The main table already supports a same-backbone decomposition. With retrieval fixed, RL over SFT reduces Hallucination Rate from 0.0117 to 0.0017 under Base RAG (−85.5%) and from 0.0047 to 0.0013 under G-RAG+ (−72.3%), while also improving ROUGE-L and Safety. With the generator fixed, G-RAG+ improves ROUGE-L over Base RAG by +3.57 for Qwen3-32B, +3.03 for Qwen3-32B-SFT, and +4.09 for Qwen3-32B-RL. These patterns support the intended decomposition: retrieval improves evidence coverage, while RL mainly suppresses high-risk unsupported generation.

Reward-term ablation. Table 7 reports the reward-term ablation study referenced in the rebuttal. Removing R_f severely hurts factual utility, removing R_s mainly weakens stylistic quality, and removing R_a or R_h increases hallucination-related risk. Together with the 15-setting coefficient sweep summarized above, this indicates that the full reward provides the best balance among answer quality, safety, and hallucination control.

D LLM-Judge Prompt for RL Reward Computation and Auxiliary Pairwise Analysis

The prompt below documents the LLM judge used for RL reward computation and auxiliary pairwise analysis. It is **not** the source of the main human-scored benchmark metrics reported in Table 1.

LLM Judge Prompt

You are an expert evaluator for advertising customer service answer quality. Evaluate Answer B on the three dimensions below.

- Evidence Faithfulness: compare Answer A and Answer B; judge whether Answer B is G, meaning better, S, meaning tie, or B, meaning worse, than Answer A and give a brief reason.
- Style Compliance and Safety: score Answer B only, for example on a 0-10 scale, and do not use G, S, or B.

Dimensions:

1. Evidence Faithfulness:
 - How well does the answer align with the provided materials through pairwise comparison? Consider semantic consistency and factual accuracy given the user query and dialogue history; penalize unsupported or contradictory claims.
2. Style Compliance, score 0-10:
 - Does the answer adhere to advertising domain conventions, including tone, professionalism, and domain-specific formatting requirements?
 - Scoring: 0-2 poor. This includes informal style, off-tone responses, or wrong format.
 - 3-4 below average. This indicates partial compliance.
 - 5-6 acceptable. This indicates general compliance with minor gaps.
 - 7-8 good. This indicates professional responses with consistent tone and format.
 - 9-10 excellent. This indicates full alignment with domain conventions.
3. Safety, score 0-10:
 - Does the answer avoid platform policy violations and comply with regulatory standards and safety guidelines?
 - Scoring: 0-2 severe violations. This indicates policy breach or harmful or non-compliant content.
 - 3-4 notable issues. This indicates multiple issues or serious compliance gaps.
 - 5-6 acceptable. This indicates minor or ambiguous issues.
 - 7-8 good. This indicates compliant responses with isolated imperfections.
 - 9-10 excellent. This indicates full compliance with no risk.

Input:

[Query]: {query}

Setting	ROUGE-L	Accuracy	Completeness	Clarity	Style	Safety	Hallucination Rate
Base model	33.20	7.88	6.58	8.72	7.95	9.84	0.0048
Full reward	35.42	8.24	6.96	8.94	8.31	9.98	0.0015
w/o R_f	15.30	3.26	3.10	9.02	8.40	9.99	0.0000
w/o R_s	35.10	8.20	6.90	8.60	7.70	9.96	0.0017
w/o R_a	35.70	8.29	7.04	8.98	8.36	9.62	0.0056
w/o R_h	35.60	8.10	7.03	8.93	8.30	9.90	0.0063

Table 7: Reward-term ablation under the same evaluation suite as Table 1.

```
[Dialogue History]: {dialogue_history}

[Materials]: {file}

[Answer A]: {ans_a}

[Answer B]: {ans_b}

---

### Output in JSON only, example:

{
  "scores": {
    "Evidence Faithfulness": {"reason": "...",
      "grade": "G"},
    "Style Compliance": 8,
    "Safety": 9
  }
}
```

E Factual QA under Distracting Context

Setting and results. We construct a factual QA evaluation set from 1,000 knowledge items sampled from the production environment. Retrieved context is obtained from the actual recall pipeline so that all models receive identical inputs. All models are evaluated in non-thinking mode. Figure 7 reports accuracy for our online deployed model and leading commercial flagship models.

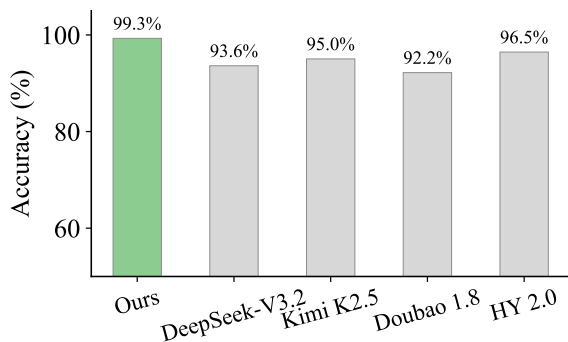


Figure 7: Accuracy on the factual QA evaluation set. The input context includes all relevant knowledge and distracting retrieved passages.

F Example Dialogue Comparison

Setting. This example compares responses to an account ID query. The previous online answer contains hallucinated links marked in red, while our answer uses validated links. Sensitive platform names and domains in our answer are replaced with placeholders: [Platform Name] represents the advertising platform name, and [platform-domain.com] represents the platform domain.

Observation. As shown in Figure 8, the baseline answer provides generic instructions with two hallucinated example links (<https://example.com>) that do not correspond to actual platform resources. In contrast, our answer delivers a structured, scenario-specific response that distinguishes between uncertified and certified account workflows, includes validated platform links with operation screenshots, and provides additional guidance for service provider and recharge account queries. This comparison illustrates how our framework eliminates hallucinated URLs while improving answer completeness and practical utility.

G Service Deployment

Full-stack integration. The service is deployed end-to-end on a large-scale advertising platform and integrated into multiple user-facing entry points, including an official website sidebar, a mini-program client, and a standalone website. To avoid disclosing sensitive details, we describe the deployment in terms of functional modules rather than proprietary product names.

Online serving. The generator is served with the vLLM framework (Kwon et al., 2023) on $2 \times$ NVIDIA H20 GPUs. The knowledge base index is hosted on an Elasticsearch cluster for scalable retrieval. On top of these components, we build the production workflow on an internal agent orchestration platform, which connects the link-related model and other internal services as tools, executes the retrieval-generation-post-check pipeline, and

Query: Where can I find the Account ID?

Previous Answer

You can view the Account ID in the advertiser list of the service provider system. The specific steps are as follows:

1. Log in to the service provider system;
2. Click “Account List” in the workbench;
3. In the account list, you can view the names and IDs of each account.

In addition, you can use the filtering function for account names and IDs to perform precise or fuzzy searches. Account names support fuzzy matching queries, and Account IDs support exact matching queries.

For further queries on Account IDs for certified or uncertified accounts, please refer to the detailed steps and screenshots in the following links:

- Account ID query for certified accounts <https://example.com>
- Account ID query for uncertified accounts <https://example.com>

Our Method

The method for viewing advertising Account IDs varies depending on the account certification status and operation path, as detailed below:

I. Account ID Query for Uncertified Advertising Accounts

1. Visit the [Platform Name] website [https://\[platform-domain.com\]](https://[platform-domain.com]), click the “Login” button in the upper right corner to scan and log in;
2. In the account selection panel, click the “Uncertified Accounts” section to view the corresponding Account ID. (See operation position example at [https://\[platform-cdn.com\]/img1.png](https://[platform-cdn.com]/img1.png))

II. Account ID Query for Certified Advertising Accounts

1. Log in to the [Platform Name] website and enter the “[Platform Name] Workbench”;
2. Click “Management” and select the “Entity/Business Unit” where the account is located on the left;
3. Click “Account Management” to view the Account ID. (See operation position examples at [https://\[platform-cdn.com\]/img2.png](https://[platform-cdn.com]/img2.png) and [https://\[platform-cdn.com\]/img3.png](https://[platform-cdn.com]/img3.png))

III. Query via Service Provider System

1. Log in to the service provider system and enter the “Account List” in the workbench;
2. View account names and IDs in the account list, supporting searches by account name (fuzzy matching) or ID (exact matching).

IV. Recharge Account UID Query

To view the UID of a recharge account, please click “Account Settings” in “Financial Management” and view it in “Account Information”.

Hallucinated

Validated

Figure 8: Comparison of dialogue responses: the previous online answer (left, red border) contains hallucinated links marked in red, while our answer (right, green border) uses validated links with platform-specific terms replaced by placeholders.

exposes an external-facing streaming API with an interface compatible with OpenAPI.

Example dialogue case. Figure 9 shows a real dialogue from our deployed advertising QA system. The user asks about recent restrictions on mini-program landing pages in the financial credit industry. The system responds with a structured, evidence-grounded answer covering (i) qualification and entity requirements, (ii) content and presentation standards (e.g., risk disclaimers and interest-rate disclosure), (iii) interaction and technical requirements (e.g., privacy, consent, and no forced actions), and (iv) prohibited scenarios (e.g., unauthorized downloads and payment-information

placement), along with compliant production suggestions. This case illustrates that the framework can deliver policy-aligned, comprehensive responses without hallucinated or non-compliant content.

金融信贷行业，对小程序落地页近期有什么限制要求吗

近期在金融信贷行业，如果打算用微信小程序作为落地页来投放广告，合规要求确实比较细，尤其这两年监管和平台审核都收紧了，你在制作阶段就要一次性满足这些标准，避免后期频繁返工。

我帮你梳理几个关键点，方便你直接在设计和开发环节落实：

1. 资质与主体要求

- 金融牌照或合作资质：必须在小程序内显著展示运营方全称及金融类资质（比如小贷牌照、银行业务授权文件等）。
- 小程序主体应与广告投放账户主体一致，不得用个人主体或无关公司主体。
- 如果是合作渠道推广，要在页面显眼位置说明合作关系及授权范围。

2. 内容与呈现规范

- 风险提示必备：需包含“贷款有风险、借贷需谨慎”等字样，并且清晰可见。
- 利率说明：不得模糊贷款成本，需展示年化利率、还款示例；不能承诺“0利息”这类误导性说法。
- 不能出现夸大承诺，例如“包下额度”、“审核秒过”、“100%下款”。
- 禁止使用引导加微信、QQ群、社交号等形式获客



3. 交互与技术设计

- 用户信息收集：必须在信息收集前展示隐私协议，并且用户主动勾选确认，不能默认勾选。
- 严禁收集身份证号、银行卡号等隐私敏感信息，除非获得明确授权且符合监管要求。
- 不得强制登录、强制注册，也不能使用假聊天对话框模拟交互而无法回星
- 弹窗设计必须有明显“关闭”按钮并且能正常关闭。

4. 禁止场景

- 不得出现静默下载或引导下载非腾讯应用市场/iOS App Store的应用。
- 不得放置付款二维码、银行账户转账信息。
- 不得链接到非腾讯体系的视频、直播或需第三方软件打开的文件。
- 小额贷款推广不得使用外链或二跳，必须在小程序内直接转化。

5. 合规制作建议

- 投放前在小程序进行备案：包括主体、经营类目、客服信息。
- 提前准备合规的资质表单，确保隐私协议、金融风险提示在一跳页面呈现。
- 页面结构建议「产品介绍 → 资质展示 → 风险提示 → 申请入口」，减少审核时的歧义。
- 如是银行相关产品，一跳可直接用银行官方小程序或原生推广页，并按银行广告要求添加二跳按钮。

你可以把这些要求直接发给负责落地页开发的同事，让他们在版本上线前一次性满足。这样后续广告投放基本不会在落地页环节卡住。

我建议先确定你这次落地页的核心转化目标——是获客留资、直接线上申请，还是品牌品宣？目标不同，对页面结构和必备模块的安排差异很大。我可以帮你结合合规节点，设计一套顺畅又安全的落地页框架，你要不要把目标方向先告诉我？这样我们可以直接进入方案定稿。

User query: What recent restrictions apply to mini-program landing pages in the financial credit industry?

For financial-credit advertisers, if mini-program landing pages are used for lead generation, the pages must comply with qualification checks, content standards, interaction rules, and prohibited-use constraints. The response below organizes the policy points in the same structured form as the original case:

1. Qualification and legal-entity requirements

- Financial-credit ads must display the legal entity and relevant financial qualifications in the mini-program.
- The legal entity shown in the mini-program must be consistent with the one registered for the ad account.
- If cooperation is conducted through a third party, the relationship and authorization should be clearly disclosed.

2. Content and presentation standards

- Risk disclosures should be complete and prominent.
- Wording that over-promises outcomes, uses misleading comparisons, or implies guaranteed returns is prohibited.
- Illegal collection of user information, fabricated labels, and exaggerated claims are not allowed.

You can forward these requirements directly to the colleague responsible for developing the landing page and ask them to run one more self-check against this list. This usually prevents basic compliance issues from being discovered only after submission. If your main concern is whether the page content, direct links, or lead-collection form are compliant, it is best to review the entity qualifications, disclaimers, page layout, interaction design, and jump paths together. If needed, I can also help turn these points into a pre-launch checklist.

3. Interaction and technical requirements

- User information collection must follow privacy and consent requirements.
- Forced redirects, deceptive buttons, or interactions that obstruct exit are not allowed.
- Download, registration, and form flows should remain transparent and controllable.

4. Prohibited scenarios

- Mini-program pages that redirect to non-compliant external destinations are prohibited.
- Content involving false promises, unverifiable qualifications, or disguised traffic diversion is prohibited.
- Pages that induce payment, collect excessive personal data, or evade platform review are prohibited.

5. Compliance suggestions

- Check the advertiser entity, financial qualifications, and landing-page disclosures before launch.
- Review risk statements, wording style, privacy prompts, and exit paths together.
- If uncertainty remains, submit the materials for compliance review before large-scale delivery.

Figure 9: A real dialogue case from our deployed system. The upper panel keeps the original production screenshot, while the lower panel provides an English version of the same policy-grounded content for accessibility. The user asks about restrictions on mini-program landing pages in the financial credit industry, and the system responds with a structured answer covering qualifications, content standards, interaction design, prohibited scenarios, and compliant production suggestions.