LERA: LLM-Enhanced RAG for Ad Auction in Generative Chatbots

Haoran Sun[1], Xinrui Song[3], Xinyu Zhang[2], Zhaohua Chen[1], Xu Chu[1], Zhilin Zhang[2], Chuan Yu[2], Jian Xu[2], Bo Zheng[2], Xiaotie Deng[1]
[1]*Peking University* [2]*Alibaba Group* [3]*Shandong University of Science and Technology*
SUNHAORAN0301@STU.PKU.EDU.CN
{CHENZHAOHUA, CHU_XU, XIAOTIE}@PKU.EDU.CN
{ZXY479640, ZHANGZHILIN.PT, YUCHUAN.YC, XIYU.XJ, BOZHENG}@ALIBABA-INC.COM

Abstract

The integration of advertising auction mechanisms into large language model (LLM)-based chatbots presents a significant opportunity for commercialization, yet poses unique challenges in balancing relevance, efficiency, and user experience. Recently, Feizi et al. [6] and Hajiaghayi et al. [8] outlined a retrieve-then-generate paradigm that decouples retrieval and generation, offering lightweight ad insertion and payment determination. However, current retrieval relies solely on text embedding similarity, which may lead to commercial misinterpretation and issues such as repetitive insertions. In this paper, we propose LERA, a two-stage retrieve-then-generate auction framework tailored for LLM chatbots. In the first stage, embedding-based coarse filtering pre-selects a small set of candidate advertisers. In the second stage, the LLM itself is queried with a designed prompt to produce logits over candidates, which serve as refined organic relevance scores. These scores are combined with bids, and a critical-value payment rule accounts for both the coarse-filtering and fine-ranking thresholds, ensuring truthfulness for utility-maximizing advertisers. The framework naturally extends to multiple ad insertions within dynamic dialogue flows and long responses. Experiments on a synthetic advertiser-query benchmark show that LERA substantially improves ad selection accuracy and insertion diversity while incurring controllable latency overhead.

1. Introduction

Large language models (LLMs) are increasingly becoming conversational gateways through which users access information, products, and services [3, 23, 24]. This shift creates new opportunities for monetizing generative systems beyond subscription and token-based pricing, but it also changes the nature of advertising auctions. Unlike sponsored search or display advertising, where ads are placed in predefined slots, chatbot advertising must decide whether an advertisement should appear at all and how it should be woven into a generated response. As a result, the auction problem becomes inseparable from language understanding, generation quality, and user experience [6, 22].

Existing designs for generative advertising can be broadly viewed through two paradigms. Retrieve-then-generate methods, such as the RAG-based auction of Hajiaghayi et al. [8], insert ads before generation and are therefore computationally lightweight. However, their allocation decisions typically rely on embedding similarity, which can miss commercial intent, negation, and intent shifts. Generate-then-retrieve or placeholder-based approaches use richer LLM signals after or during generation [1, 4, 17, 21, 26], but often require multiple LLM calls, advertiser-specific scoring, or explicit slots that may hurt fluency. A fuller comparison with LLM-mediated advertising, RAG, recommendation, and externality-aware auctions is provided in Appendix B.

© Sun et al..

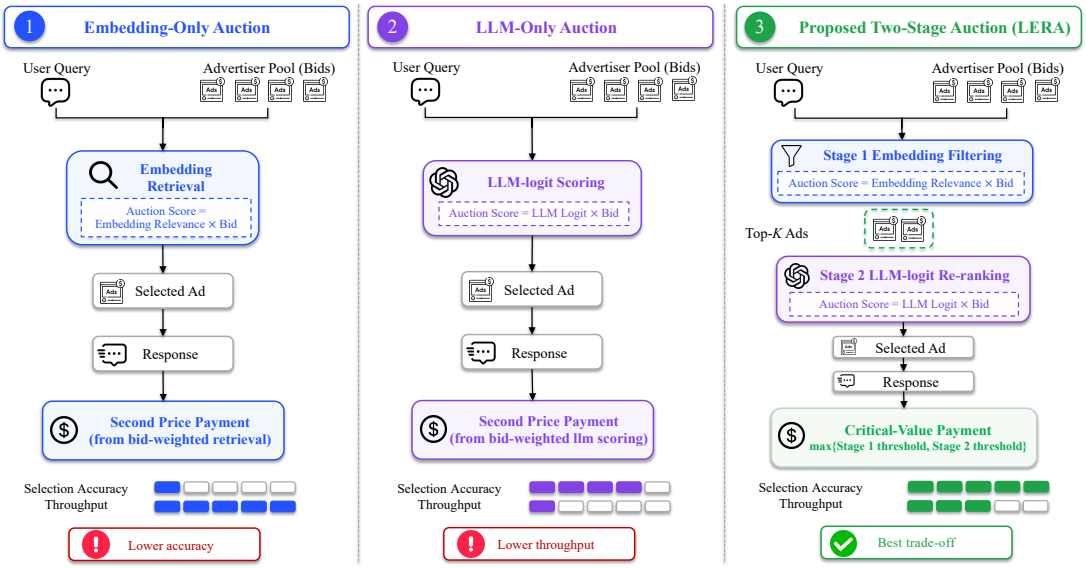

Figure 1: Comparison of ad selection strategies for LLM-based chatbots. **Embedding-only** uses embedding similarity weighted by bids; **LLM-only** scores all advertisers with LLM logits; **LERA** first filters candidates with embedding similarity and then applies LLM-logit scoring on the top-K set. Payment annotations indicate the auction mechanism, while the bars illustrate relative selection accuracy and system throughput.

Directly using an LLM as the ad selector is also not by itself a complete solution. LLMs have strong contextual understanding and have been used for ranking and recommendation [11, 18, 27], but incorporating their judgments into an auction requires a payment rule compatible with strategic bidding. Moreover, scoring all advertisers with LLM logits is expensive under high-throughput inference engines: systems such as SGLang and vLLM are optimized for generation, while candidate-level log-probability extraction often requires parallel prefilling over the candidate set. Thus, a practical chatbot auction must use LLM judgment selectively rather than exhaustively.

We propose LERA, a two-stage retrieve-then-generate auction framework for ad insertion in LLM-based chatbots. As shown in Figure 1, LERA first applies embedding-based coarse filtering to preselect a small set of candidate advertisers from a large pool. It then presents these candidates, together with a no-insertion option, to the LLM and extracts logits over candidate labels as refined organic relevance scores. These scores are combined with bids to determine the winner, while a critical-value payment rule accounts for both the coarse-filtering threshold and the fine-ranking threshold. In this way, LERA keeps the efficiency of retrieval-based allocation while using the LLM's contextual judgment where it is most valuable.

Our contributions are threefold. First, we introduce a context-aware ad auction framework that combines embedding-based retrieval with LLM-logit-based fine ranking. Second, we derive a critical-value payment rule tailored to the two-stage allocation process. Third, experiments on single-ad and multi-ad insertion tasks show that LERA better captures nuanced commercial intent, improves insertion diversity, and achieves a favorable accuracy–efficiency trade-off compared with embedding-only and LLM-only alternatives.

2. Method: Two-Stage Ad Auction

Problem setup. Given a user query x, N advertisers bid for the opportunity to have their product integrated into the response. Advertiser i provides a public description d_i and a reported bid b_i, while its value-per-click v_i is private. The mechanism maps $(x, \{d_i\}_{i=1}^N, \{b_i\}_{i=1}^N)$ to an allocation, i.e., a generated response y, and a payment vector. In retrieve-then-generate auctions [8], allocation is usually decoupled into a retrieval decision followed by response generation: an advertiser is selected according to a bid-weighted relevance score, and the selected ad is then inserted into the prompt. This design is efficient, but it places heavy pressure on the relevance score, since embedding similarity alone may treat rejected products, initial contextual mentions, or near-duplicate categories as positive evidence.

Stage 1: coarse candidate filtering. LERA keeps the retrieve-then-generate structure but replaces the single relevance score with a two-stage score. Rather than directly embedding the raw query, we first prompt the LLM to generate a concise set of intent keywords, denoted by $\mathbf{w} \sim \text{GenAI}(\text{Prompt}_{\text{keywords}}(x))$. For each advertiser, we compute

$$s_i^{(1)} = \text{Norm}\big(\text{Similarity}(\text{Emb}(\mathbf{w}), \text{Emb}(d_i))\big), \qquad \text{Score}_i^{(1)} = s_i^{(1)} b_i.$$

The top-K advertisers according to $\text{Score}_i^{(1)}$ form the candidate set $\mathcal{S}$. This stage is inexpensive and reduces the scoring problem from the full advertiser pool to a small set where $K \ll N$.

Stage 2: LLM-logit fine ranking. The second stage refines the relevance assessment within $\mathcal{S}$. We present the original query and candidate descriptions to the LLM, include a no-insertion option ϕ, and define $\hat{\mathcal{S}} = \mathcal{S} \cup \{\phi\}$. Instead of relying on generated text, we extract the model's logits $\mathbf{z}$ over candidate labels from $\text{Prompt}_{\text{ranking}}(x, \{d_j\}_{j \in \hat{\mathcal{S}}})$ and normalize them as

$$s_j^{(2)} = \frac{\exp(z_j)}{\sum_{k \in \hat{\mathcal{S}}} \exp(z_k)}.$$

This score is interpreted as the LLM's refined organic relevance within the shortlisted set. The final auction score is $\text{Score}_j^{(2)} = s_j^{(2)} b_j$, with $b_\phi = 1$ for the no-insertion option, and the winner is

$$i^* = \arg\max_{j \in \hat{\mathcal{S}}} \text{Score}_j^{(2)}.$$

If $i^* = \phi$, the response is generated without an ad; otherwise the winning advertiser description is incorporated into the final prompt. This preserves the single full-response generation call while adding only a small LLM scoring step over the candidate set.

Payment rule. The two-stage allocation creates two thresholds: the advertiser must enter the Stage 1 top-K set and then win the Stage 2 ranking. Following critical-value pricing [9, 19], if $i^* \neq \phi$, let $\text{Score}_{(K+1)}^{(1)}$ be the Stage 1 entry threshold and $\text{Score}_{(2)}^{(2)}$ be the second-highest Stage 2 score in $\hat{\mathcal{S}}$. We charge

$$p_{i^*} = \max\left\{ \frac{\text{Score}_{(K+1)}^{(1)}}{s_{i^*}^{(1)}}, \frac{\text{Score}_{(2)}^{(2)}}{s_{i^*}^{(2)}} \right\}.$$

The maximum is necessary because satisfying only one threshold is insufficient: a bidder that cannot enter the shortlist cannot be selected, and a bidder that enters but cannot beat the second-stage competitor also loses.

Proposition 1 *Assume that the relevance scores used by the allocation rule are independent of advertiser i's reported bid except through the bid-weighted ranking rule. Further assume that the click outcome $o_i(y)$ and ex-post value $\hat{v}_i(y)$ are independent conditioned on the ad being displayed, and that $v_i = \mathbb{E}_y[\hat{v}_i(y)]$. Then the two-stage auction mechanism with the payment rule above is incentive compatible for a utility-maximizing advertiser.*

The proof follows the standard critical-value argument: under the stated assumptions, expected utility reduces to $(v_i - p_i)\bar{o}_i$ with $\bar{o}_i \geq 0$, and the payment is exactly the minimum bid required to satisfy both allocation thresholds. The full proof and discussion of the independence assumption are deferred to Appendix G.

Multiple insertions. The same mechanism extends to long-form or multi-turn responses by dividing generation into T segments. For segment t, the auction is conditioned on the original query x and response prefix $y_{<t}$, selects either an advertiser i_t^* or ϕ, and then generates the next segment with the selected advertiser omitted when $i_t^* = \phi$. The main additional costs are prefix reprocessing, keyword generation, and Stage 2 logit computation. Because T and K are small in our setting, the overhead remains controllable, as evaluated in Section 3.2.

3. Experimental Results

We evaluate LERA against embedding-only retrieval and several ablations. All experiments use a synthetic advertiser benchmark with 100 advertisers from 8 product categories; each advertiser has a short brand description used for both retrieval and LLM scoring. We use uniform bids ($b_i = 1.0$) to isolate allocation quality from heterogeneous bidding. The Gen Model is used for keyword generation, logit scoring, and final response generation, and we evaluate 4B, 30B, and 235B model scales served by SGLang [31]. Dataset examples, prompts, and implementation details are provided in Appendices D–E.

3.1. Single Ad Selection Accuracy

Single ad insertion is the elementary allocation task: given a user query, the mechanism must select the most suitable advertiser for insertion. We construct 240 test queries with unique ground-truth advertisers, covering three difficult intent scenarios: complex implicit constraints, natural intent shifts, and explicit rejection of superficially similar products. These cases are designed to test whether a method can distinguish commercial intent from mere semantic overlap.

Table 1 shows that embedding-only retrieval performs poorly across all categories, even with an 8B embedding model. This is expected because dense retrieval is trained primarily for semantic proximity, while ad selection often requires reasoning about negation, changing intent, and the suitability of an ad inside a generated response. In contrast, LERA consistently improves selection accuracy, reaching 94.6%, 98.73%, and 97.5% overall accuracy with 4B, 30B, and 235B Gen Models, respectively.

The gains are especially clear for smaller models. With the 4B and 30B Gen Models, LERA outperforms both keyword-only retrieval and direct LLM-only scoring over all advertisers. This

Table 1: **Accuracy of single-ad insertion across varying strategies and model scales.** We evaluate the selection accuracy of Embed Models and Gen Models on three distinct simulated scenarios: Complex Intent Analysis, Natural Intent Shift, and Negative Product Refinement. LERA consistently outperforms the embedding-only baseline.

Method	Embed Model	Gen Model	Complex	Shift	Negative	Overall
Embedding-only	all-MiniLM-L6-v2	N/A	67.5%	68.8%	52.5%	62.93%
	Qwen3-Embedding-8B	N/A	55.0%	60.0%	53.8%	56.3%
LERA (LLM-only)	N/A	Qwen3-4B	76.2%	71.2%	72.5%	73.3%
	N/A	Qwen3-30B-A3B	97.5%	86.2%	100%	94.57%
	N/A	Qwen3-235B-A22B	100%	100%	100%	100%
LERA (Keyword-Match)	all-MiniLM-L6-v2	Qwen3-4B	67.5%	73.8%	82.5%	74.6%
	all-MiniLM-L6-v2	Qwen3-30B-A3B	81.2%	90.0%	88.8%	86.67%
	all-MiniLM-L6-v2	Qwen3-235B-A22B	75.0%	81.2%	86.2%	80.8%
LERA	all-MiniLM-L6-v2	Qwen3-4B	91.2%	97.5%	95.0%	94.6%
	all-MiniLM-L6-v2	Qwen3-30B-A3B	96.2%	100%	100%	98.73%
	all-MiniLM-L6-v2	Qwen3-235B-A22B	96.2%	98.8%	97.5%	97.5%

suggests that coarse filtering is not merely an efficiency device: it also simplifies the fine-ranking problem by presenting the LLM with a smaller, more relevant candidate set. The LLM-only variant reaches perfect accuracy with the 235B model, but this comes with a substantial systems cost because all advertisers must be scored through parallel prefilling. We discuss this implementation issue in Appendix C.2 and quantify the throughput cost below.

3.2. Dynamic Multi-Ad Insertion

We next evaluate a more realistic dynamic setting where a response may contain multiple advertisements. We construct 50 queries in which users request recommendations from three distinct product categories. Following the retrieve-then-generate workflow, response generation is split into segments; before each segment, the auction decides whether to insert an ad and which advertiser to select. We limit the maximum number of inserted ads to three, use $K = 8$ for dynamic insertion, and measure query satisfaction as the average proportion of requested categories covered by inserted ads.

Table 2: **Query satisfaction in dynamic multi-ad insertion scenarios.** The metric represents the average proportion of user-requested categories successfully fulfilled. LERA achieves the best balance of context-aware diversity compared to ablation variants.

Method	4B	30B-A3B	235B-A22B
Embedding-only	46.67%	42.00%	45.33%
LERA (LLM-only)	48.61%	64.00%	86.67%
LERA (Keyword-Match)	64.67%	63.33%	64.67%
LERA (Cosine-LLM)	63.33%	58.67%	58.00%
LERA	72.00%	80.00%	80.67%

As shown in Table 2, LERA improves over the Embedding-only baseline by a large margin across all model scales. The keyword-only and cosine-filtering ablations also improve over the baseline, but they remain substantially below LERA, indicating that effective dynamic insertion requires both query-aware coarse filtering and second-stage LLM-logit ranking. LLM-only scoring improves sharply with model scale and achieves the best satisfaction with the 235B model, but its

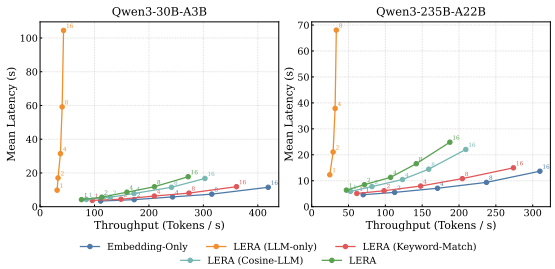

Figure 2: **Latency and throughput tradeoff.** Comparison of LERA and its ablation variants under different concurrency levels using 30B and 235B Gen Models.

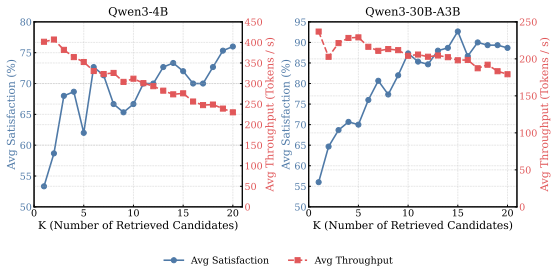

Figure 3: **Effect of first-stage winner set size K.** Increasing K generally improves allocation quality but reduces throughput.

accuracy-oriented advantage must be weighed against the inference overhead of scoring the entire advertiser pool.

Efficiency. Figure 2 compares latency and throughput under different concurrency levels. LERA (LLM-only) quickly reaches a throughput bottleneck because SGLang computes logits through parallel prefilling over a large candidate set. LERA incurs moderate overhead relative to the Embedding-only baseline, especially at low concurrency, while avoiding the much larger cost of scoring all advertisers. This demonstrates the intended tradeoff: LERA spends additional inference on a small shortlist in exchange for substantially better ad matching. In deployment scenarios, this cost can be justified when improved matching accuracy translates into higher platform revenue, better advertiser utility, or a less repetitive user experience. Designing auction mechanisms that better align with high-throughput LLM serving systems remains an important direction for future work.

Ablation Study on the First-Stage Candidate Set Size K. Finally, we study how the first-stage candidate set size K affects LERA. As shown in Figure 3, increasing K generally improves performance because the second-stage LLM scorer has access to a broader candidate set. However, once K becomes sufficiently large, the performance gain diminishes and may even slightly decrease. One possible reason is that a larger candidate set introduces less relevant advertisers into the second-stage ranking problem, making fine-grained LLM scoring more difficult. In terms of efficiency, throughput decreases as K increases because more candidates require LLM-based scoring. These results suggest that K should be chosen to balance allocation quality and inference efficiency, rather than being set as large as possible.

4. Conclusion

We presented LERA, a two-stage auction mechanism for advertisement insertion in LLM-based chatbots. By combining efficient embedding-based filtering with LLM-logit-based fine ranking, LERA captures commercial intent that pure embedding retrieval often misses while avoiding the high cost of scoring every advertiser with an LLM. Its critical-value payment rule accounts for both shortlist entry and final-stage competition, providing an incentive-compatible mechanism under standard assumptions. Experiments show that LERA improves single-ad selection accuracy and dynamic multi-ad category coverage with a controllable efficiency tradeoff, while the main limitations of synthetic evaluation and the independence assumption are discussed in Appendix H.

References

[1] Santiago Balseiro, Kshipra Bhawalkar, Yuan Deng, Zhe Feng, Jieming Mao, Aranyak Mehta, Vahab Mirrokni, Renato Paes Leme, Di Wang, and Song Zuo. Position auctions in ai-generated content. *arXiv preprint arXiv:2506.03309*, 2025.

[2] Sunhao Dai, Ninglu Shao, Haiyuan Zhao, Weijie Yu, Zihua Si, Chen Xu, Zhongxiang Sun, Xiao Zhang, and Jun Xu. Uncovering chatgpt's capabilities in recommender systems. In *Proceedings of the 17th ACM Conference on Recommender Systems*, pages 1126–1132, 2023.

[3] DeepSeek-AI. Deepseek-v3.2: Pushing the frontier of open large language models, 2025.

[4] Paul Duetting, Vahab Mirrokni, Renato Paes Leme, Haifeng Xu, and Song Zuo. Mechanism design for large language models. In *Proceedings of the ACM Web Conference 2024*, pages 144–155, 2024.

[5] Benjamin Edelman, Michael Ostrovsky, and Michael Schwarz. Internet advertising and the generalized second-price auction: Selling billions of dollars worth of keywords. *American economic review*, 97(1):242–259, 2007.

[6] Soheil Feizi, MohammadTaghi Hajiaghayi, Keivan Rezaei, and Suho Shin. Online advertisements with llms: Opportunities and challenges. *arXiv preprint arXiv:2311.07601*, 2023.

[7] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yixin Dai, Jiawei Sun, Haofen Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2(1), 2023.

[8] MohammadTaghi Hajiaghayi, Sébastien Lahaie, Keivan Rezaei, and Suho Shin. Ad auctions for llms via retrieval augmented generation. *Advances in Neural Information Processing Systems*, 37:18445–18480, 2024.

[9] Paul Klemperer. Auction theory: A guide to the literature. *Journal of economic surveys*, 13 (3):227–286, 1999.

[10] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474, 2020.

[11] Fengxin Li, Yi Li, Yue Liu, Chao Zhou, Yuan Wang, Xiaoxiang Deng, Wei Xue, Dapeng Liu, Lei Xiao, Haijie Gu, et al. Leadre: Multi-faceted knowledge enhanced llm empowered display advertisement recommender system. *arXiv preprint arXiv:2411.13789*, 2024.

[12] Lei Li, Yongfeng Zhang, Dugang Liu, and Li Chen. Large language models for generative recommendation: A survey and visionary discussions. In *Proceedings of the 2024 joint international conference on computational linguistics, language resources and evaluation (LREC-COLING 2024)*, pages 10146–10159, 2024.

[13] Ningyuan Li, Yunxuan Ma, Yang Zhao, Zhijian Duan, Yurong Chen, Zhilin Zhang, Jian Xu, Bo Zheng, and Xiaotie Deng. Learning-based ad auction design with externalities: the framework and a matching-based approach. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1291–1302, 2023.

[14] Xuejian Li, Ze Wang, Bingqi Zhu, Fei He, Yongkang Wang, and Xingxing Wang. Deep automated mechanism design for integrating ad auction and allocation in feed. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1211–1220, 2024.

[15] Junling Liu, Chao Liu, Peilin Zhou, Renjie Lv, Kang Zhou, and Yan Zhang. Is chatgpt a good recommender? a preliminary study. *arXiv preprint arXiv:2304.10149*, 2023.

[16] Xiangyu Liu, Chuan Yu, Zhilin Zhang, Zhenzhe Zheng, Yu Rong, Hongtao Lv, Da Huo, Yiqing Wang, Dagui Chen, Jian Xu, et al. Neural auction: End-to-end learning of auction mechanisms for e-commerce advertising. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 3354–3364, 2021.

[17] Tommy Mordo, Moshe Tennenholtz, and Oren Kurland. Sponsored question answering. In *Proceedings of the 2024 ACM SIGIR International Conference on Theory of Information Retrieval*, pages 167–173, 2024.

[18] Martin Reisenbichler, Thomas Reutterer, and David A Schweidel. Applying large language models to sponsored search advertising. *Marketing Science*, 2025.

[19] John G Riley and William F Samuelson. Optimal auctions. *The American Economic Review*, 71(3):381–392, 1981.

[20] Tolga Şakar and Hakan Emekci. Maximizing rag efficiency: A comparative analysis of rag methods. *Natural Language Processing*, 31(1):1–25, 2025.

[21] Ermis Soumalias, Michael J Curry, and Sven Seuken. Truthful aggregation of llms with an application to online advertising. *arXiv preprint arXiv:2405.05905*, 2024.

[22] Brian Jay Tang, Kaiwen Sun, Noah T Curran, Florian Schaub, and Kang G Shin. Genai advertising: Risks of personalizing ads with llms. *arXiv preprint arXiv:2409.15436*, 2024.

[23] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

[24] Qwen Team. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

[25] Hal R Varian. Position auctions. *international Journal of industrial Organization*, 25(6): 1163–1178, 2007.

[26] Shengwei Xu, Zhaohua Chen, Xiaotie Deng, Zhiyi Huang, and Grant Schoenebeck. Ad insertion in llm-generated responses. *arXiv preprint arXiv:2601.19435*, 2026.

[27] Qi Yang, Marlo Ongpin, Sergey Nikolenko, Alfred Huang, and Aleksandr Farseev. Against opacity: Explainable ai and large language models for effective digital advertising. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 9299–9305, 2023.

[28] Zhilin Zhang, Xiangyu Liu, Zhenzhe Zheng, Chenrui Zhang, Miao Xu, Junwei Pan, Chuan Yu, Fan Wu, Jian Xu, and Kun Gai. Optimizing multiple performance metrics with deep gsp auctions for e-commerce advertising. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 993–1001, 2021.

[29] Chujie Zhao, Qun Hu, Shiping Song, Dagui Chen, Han Zhu, Jian Xu, and Bo Zheng. Llm-auction: Generative auction towards llm-native advertising. *arXiv preprint arXiv:2512.10551*, 2025.

[30] Zihuai Zhao, Wenqi Fan, Jiatong Li, Yunqing Liu, Xiaowei Mei, Yiqi Wang, Zhen Wen, Fei Wang, Xiangyu Zhao, Jiliang Tang, et al. Recommender systems in the era of large language models (llms). *IEEE Transactions on Knowledge and Data Engineering*, 36(11):6889–6907, 2024.

[31] Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue Sun, Jeff Huang, Cody H Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph E Gonzalez, et al. Sglang: Efficient execution of structured language model programs. *Advances in neural information processing systems*, 37: 62557–62583, 2024.

[32] Ruitao Zhu, Yangsu Liu, Dagui Chen, Zhenjia Ma, Chufeng Shi, Zhenzhe Zheng, Jie Zhang, Jian Xu, Bo Zheng, and Fan Wu. Contextual generative auction with permutation-level externalities for online advertising. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pages 2171–2181, 2025.

Appendix A. Appendix Overview

This appendix provides additional materials supporting the main paper. Appendix B expands the discussion of related work. Appendix C reports implementation details and generation settings for reproducibility. Appendix D describes the construction of the synthetic advertiser-query benchmark. Appendix E lists the prompt templates used in different stages of the framework. Appendix F provides representative case studies, Appendix G contains the omitted proof of Proposition 1, and Appendix H discusses limitations and broader impacts.

Appendix B. Additional Related Work

This section provides a more detailed discussion of the literature related to our work. Our framework lies at the intersection of LLM-mediated advertising, retrieval-augmented generation, LLM-based recommendation, and externality-aware auction design. We therefore organize the discussion around four themes: ad auctions in LLM interfaces, retrieval-augmented generation, ad recommendation with LLMs, and externalities in auction mechanisms.

B.1. Ad Auctions in LLMs

Research on integrating advertising auctions into LLM-based chatbots is a rapidly emerging area. Existing frameworks can generally be categorized into *generate-then-retrieve* and *retrieve-then-generate* paradigms, depending on whether the auction is applied after a candidate response is generated or before the final response is produced.

In the *generate-then-retrieve*, or post-generation, approach, the model first generates candidate responses or probability distributions, which are then used as the basis for auction decisions. Duetting et al. [4] pioneered a mechanism in which advertisers bid on token probabilities, aggregating preferences from multiple LLMs to produce a joint output. Similarly, Mordo et al. [17] and Soumalias et al. [21] explore auctioning the "right to speak" or inserting ads into generated placeholders. While these methods are theoretically grounded, they often require multiple inference passes or complex output aggregation, leading to high latency and inference costs [22].

In contrast, the *retrieve-then-generate* paradigm, recently formalized by Hajiaghayi et al. [8], treats advertisers as retrievable documents within a RAG framework. An auction determines which ads are retrieved and inserted into the prompt context before generation begins. This approach is modular and computationally efficient because it decouples the auction stage from the expensive generation process. However, existing implementations often rely heavily on standard retrieval scores, which may not fully align with commercial objectives. This limitation motivates our framework, which introduces a selection mechanism that explicitly combines coarse retrieval with LLM-based ad relevance estimation.

More recently, Zhao et al. [29] proposed a training method that enables an LLM to generate responses satisfying bid monotonicity. Although this approach combines retrieval and generation, it relies on costly model training and may be less flexible when the ad inventory changes frequently. Our work instead focuses on a modular inference-time mechanism that can be applied without retraining the underlying LLM.

B.2. Retrieval-Augmented Generation

Retrieval-augmented generation was introduced to ground LLM responses in external and up-to-date knowledge [10]. A standard RAG pipeline retrieves documents based on vector similarity, often using dense embeddings, and then feeds the retrieved documents into the generator [7]. This paradigm has been widely adapted for downstream tasks, including recommendation systems where items are retrieved as contextual evidence [20].

As discussed in the main paper, RAG offers a natural abstraction for ad insertion: advertisements can be viewed as external knowledge chunks that may be inserted into the LLM context. However, a fundamental difference remains between ordinary RAG and sponsored generation. Standard RAG primarily aims to maximize semantic fidelity to the user query, whereas ad insertion must also account for revenue, advertiser value, and the likelihood that an ad can be integrated without degrading user experience. In this sense, the relevant retrieval signal is not merely semantic similarity, but a broader notion of commercial compatibility. Our two-stage retrieval and scoring process adapts standard RAG principles to this competitive and revenue-driven setting.

B.3. Ad Recommendation with LLMs

Beyond auction design, a growing body of work studies LLMs as ranking and recommendation engines. Recent surveys suggest that LLMs possess strong semantic understanding that can be leveraged for zero-shot item recommendation and ranking [12, 30]. Liu et al. [15] and Dai et al. [2] show that LLMs can evaluate the relevance of candidate items to a user's query and history, often outperforming traditional embedding-based methods in capturing nuanced user intent.

The key distinction between this literature and our work is the economic layer. LLM-based recommendation studies typically focus on maximizing relevance, utility, or user satisfaction, while abstracting away from strategic advertiser behavior and incentive-compatible pricing. In contrast, our framework uses the LLM's relevance assessment as an organic compatibility score within a truthful auction mechanism. This allows us to preserve the semantic strengths of LLM-based recommendation while incorporating advertiser bids and payment rules.

B.4. Externality in Ad Auction Design

Early work on sponsored search auctions, such as the Generalized Second Price auction, primarily modeled externalities as position effects, assuming that an ad's click probability depends on its slot and inherent quality [5, 25]. As advertising shifted from search pages to recommendation feeds and mixed-content environments, later work began to study richer *allocative externalities*, where an ad's value and user impact depend on neighboring content.

To address such effects, researchers have proposed deep learning-based mechanisms. *Deep GSP* [28] and *Neural Auctions* [16] use deep networks to learn non-linear ranking scores that capture inter-item dependencies while maintaining incentive compatibility through monotone allocation rules. More recent mechanisms, including WCVCG [13], *MIAA* [14], and *Contextual Generative Auctions* [32], explicitly model whole-slate effects by jointly optimizing the arrangement of ads and organic items to balance platform revenue with user experience.

While these works successfully handle externalities in static lists or feeds, LLM responses introduce a different type of *generative externality*. Unlike a fixed feed, an LLM response is a cohesive narrative in which an ad insertion can alter the text's flow, tone, specificity, and length [1]. Our

Table 3: Summary of implementation settings.

Component	Setting
LLM backbone	Qwen models, including Qwen-4B, Qwen-30B, and Qwen-235B, are used for auction decision-making and response generation.
Embedding model	`sentence-transformers/all-MiniLM-L6-v2` is used for cosine-similarity filtering and combined retrieval strategies.
Serving framework	SGLang is used to leverage RadixAttention for efficient KV-cache reuse in dynamic multi-turn insertion.
Response generation	Temperature is set to 0.7; the maximum generation length is 512 tokens for single-turn generation and 64 tokens per chunk for dynamic generation.
Auction scoring	Temperature is set to 0.0 during LLM-based auction scoring to obtain deterministic log-probabilities for Q-score computation.
Dynamic insertion	The chat-based dynamic strategy is limited to at most 10 generation chunks and at most 3 ad insertions per session.
Initial retrieval	The top-K retrieval parameter is set to $K = 5$ in the single-turn setting and $K = 8$ in the dynamic setting.

work extends externality-aware design to this generative setting by proposing a payment rule that accounts for the dynamic organic probability of an ad within a generated response.

Overall, existing work has studied auction design for LLM-generated content, LLM-based recommendation, and externalities in traditional ad allocation settings. However, these lines of work have not fully addressed the setting where advertisements are selected before or during LLM generation and where the final response itself determines the organic compatibility of an ad. Our work complements this literature by introducing a retrieval-based auction framework that combines efficient candidate selection, LLM-based relevance estimation, and an externality-aware payment rule tailored to generative responses.

Appendix C. Implementation Details

We provide implementation details to facilitate reproducibility and to clarify how the proposed mechanism is instantiated in practice. Unless otherwise stated, all methods are evaluated under the same model, retrieval, and generation settings, so that performance differences can be attributed primarily to the auction and selection strategies rather than to implementation-specific choices.

Our experiments are implemented using the SGLang framework. We use SGLang primarily to leverage its RadixAttention mechanism for efficient KV-cache reuse, which is important for our multi-turn dynamic insertion strategies. In these strategies, the system repeatedly evaluates whether a new advertisement should be inserted as the response unfolds, and therefore benefits from reusing previously computed context representations.

These settings are designed to separate the effects of retrieval and auction design from those of stochastic decoding. In particular, deterministic scoring is used for the auction stage, while moderate-temperature generation is retained for producing natural conversational responses.

C.1. Detailed Inference Hyperparameters

Table 4 provides a comprehensive list of inference hyperparameters used in our experiments, extracted from the implementation code for full reproducibility.

Table 4: Detailed inference hyperparameters.

Hyperparameter	Value / Setting
General Settings	
Device	`cuda` (NVIDIA A800 GPUs)
Random seed	42
Maximum concurrent queries	8
Response Generation	
Generation temperature	0.7
Max new tokens (single-turn)	512
Max new tokens (dynamic per chunk)	64
Sampling strategy	Top-p sampling (default SGLang)
Keyword Generation (Stage 1)	
Keyword count range	2–3 keywords
Keyword gen temperature	0.7
Keyword gen max tokens	64
LLM Auction Scoring (Stage 2)	
Scoring temperature	0.0 (deterministic)
Logprob source	SGLang `normalized_prompt_logprobs`
Choice symbols	Numeric ("1", "2", ...) or single-token when enabled
No-insertion symbol	"0"
No-insertion margin	0.0
Score normalization	Softmax over logprobs
Two-Stage Filtering	
Top-K candidates (single-turn)	$K = 5$
Top-K candidates (dynamic)	$K = 8$
Chat-Based Dynamic Strategy	
Maximum generation chunks	10 (`max_parts`)
Maximum ad insertions	3 (`max_insertions`)
System prompt	"You are a helpful assistant..." (see Appendix E)
Response termination	Empty, whitespace, or "END" token
Bid Sampling	
Alternative	Uniform 1.0 (`--all_one_bids` flag)

C.2. Logit Computation with SGLang

As introduced in Section 3, we utilize SGLang [31] for our LLM inference to reflect the performance characteristics of modern, high-throughput deployment environments. However, extracting log probabilities (log probs) for the second-stage scoring requires specific engineering considerations due to the design of such inference engines.

In the second stage, the mechanism evaluates $K + 1$ candidate choices to compute refined relevance scores. Standard inference libraries, such as Hugging Face `transformers`, allow direct access to the model's raw computational results, enabling the extraction of full vocabulary logits in a single forward pass. In contrast, highly optimized serving frameworks like SGLang prioritize overall throughput and memory management, and therefore do not natively expose intermediate logits for arbitrary token choices during a standard generation call.

To obtain the required log probabilities, SGLang evaluates multiple choices through parallel prefilling. Under this mechanism, $K+1$ distinct prompt variations are constructed—each explicitly appending one of the candidate selections. The inference engine processes these prompts concurrently and returns the sequence-level log probabilities, which are then used directly to derive the final Stage 2 auction scores.

Although the Hugging Face engine could theoretically achieve this logit extraction with fewer forward passes, its inference efficiency is significantly lower. Because most practical, large-scale LLM services today are deployed on high-throughput engines like SGLang or vLLM, choosing SGLang for our experiments ensures that the measured efficiency metrics, latency tradeoffs, and observed throughput bottlenecks accurately reflect real-world production constraints.

Appendix D. Dataset Construction Details

This section describes the construction of our synthetic benchmark for evaluating ad selection in LLM-mediated responses. The goal of the dataset is not to reproduce a specific commercial ad inventory, but to create controlled query-ad matching scenarios in which semantic relevance, commercial intent, negation, and intent shifts can be systematically evaluated.

The dataset contains advertiser descriptions and user queries, both generated with the assistance of LLMs and manually organized into controlled evaluation groups. Advertiser descriptions are designed to be short enough to fit naturally into prompts while still containing specific product attributes that allow fine-grained matching. User queries are constructed to test whether a method can distinguish true commercial intent from superficial lexical or semantic overlap.

D.1. Advertiser Categories and Descriptions

We define eight product categories and generate multiple advertiser descriptions for each category, resulting in 100 unique advertisers. Each advertiser description is approximately 20–30 words long and highlights product features, target users, or use cases. Table 5 lists the categories and a representative advertiser description for each.

D.2. Test Queries

The test queries are designed to stress different aspects of intent understanding. In addition to straightforward product matching, the benchmark includes queries involving implicit constraints, natural intent shifts, and explicit rejection of superficially similar products. These cases are particularly challenging for embedding-based retrieval methods because lexical or semantic overlap may point to products that the user explicitly does not want.

We construct two query sets: 240 single-ad queries and 50 multi-ad queries. Single-ad queries require the system to select one appropriate advertiser, while multi-ad queries require the system to identify and address multiple distinct product needs within a single response.

Table 5: Advertiser categories and representative advertiser descriptions.

Category	Representative Description
Snacks	**CrispFalls Potato Chips.** CrispFalls specializes in classic, ridged potato chips with a crisp snap, targeting traditional snackers who prefer salty crunch and familiar flavors.
Beverages	**VelvetBean Cold Brew.** VelvetBean offers nitrogen-infused cold brew coffee sourced from single-origin Ethiopian beans, designed for coffee drinkers seeking a naturally sweet and creamy texture.
Electronics	**SilentStorm Noise-Cancelling Earbuds.** SilentStorm produces true wireless earbuds with active noise cancellation for commuters and focus-seekers who want immersive audio.
Tops	**SummitShell Rain Jacket.** SummitShell designs lightweight, waterproof, and breathable rain jackets for outdoor adventurers and urban commuters.
Footwear	**TerraTread Hiking Boots.** TerraTread manufactures rugged waterproof hiking boots with deep-lug soles for backpackers and hikers on technical terrain.
Skincare	**CitraBlast Vitamin C Serum.** CitraBlast formulates stabilized Vitamin C serum for brightening and antioxidant protection, targeting users concerned with skin dullness.
Home & Living	**NimbusLight Desk Lamp.** NimbusLight designs adjustable smart desk lamps with color-temperature control for students and professionals needing glare-free lighting.
Food & Dining	**OceanPrime Seared Salmon.** OceanPrime offers ready-to-eat, sustainably sourced Atlantic salmon fillets prepared using a sous-vide technique.

D.2.1. SINGLE-AD QUERIES

Single-ad queries are stratified into three groups, with 80 queries per group. Each group targets a different form of user intent complexity.

Type 1: Complex Intent Analysis. These queries contain nuanced constraints that require understanding multiple implicit requirements, such as avoiding certain product characteristics while seeking specific qualities.

Type 1 Example: Complex Intent Analysis

User query. I'm having friends over for a movie marathon. I don't want anything greasy that will ruin the couch, nor heavy candy bars. Just looking for a light, airy bite that perfectly balances sweet and salty flavors in a classic bowl.

Ground truth. PoppyHaven Kettle Popcorn.

Type 1 Example: Complex Intent Analysis

User query. I need to hit my macros after leaving the gym, but I despise the chalky texture of typical supplements. I'm looking for something that feels like a decadent, chewy dessert but is packed with whey isolate.

Ground truth. NutRidge Protein Granola Bars.

Type 2: Natural Intent Shift. These queries initially reference products from other categories or similar use cases, then naturally transition to the target category. A successful method must identify the user's final operative intent rather than over-weighting the initial context.

Type 2 Example: Natural Intent Shift

User query. I used to rely heavily on dark roast drip coffees and high-octane tropical energy drinks to get through my shifts, but they make me too jittery now. I'm looking for a beverage that gives a calmer focus or just a sophisticated caffeine-free refreshment to sip on.

Excluded products. BlazeBrew Dark Roast Coffee; NeonFizz Energy Drink.
Target products. MatchaZen Organic Matcha Latte; SparkLeaf Botanical Tea.

Type 2 Example: Natural Intent Shift

User query. I'm trying to cut back on greasy potato chips, even those super spicy kettle-cooked ones I usually love. For my upcoming road trip, I want to pack something savory and high in protein but without all that oil, maybe some air-dried meat or spiced legumes.

Excluded products. CrispFalls Potato Chips; LavaThin Potato Chips.
Target products. SmokeTrail Beef Jerky; PulsePoint Roasted Chickpeas.

Type 3: Negative Product Refinement. These queries mention products within the same broad category but explicitly reject them before revealing the user's true preference. This setting tests whether the method can distinguish negative evidence from positive relevance signals.

Type 3 Example: Negative Product Refinement

User query. I was looking at heavy insulated winter boots with deep traction, but that is more protection than I need. I am not hiking through snow; I just want a lightweight, breathable pair of walking shoes for everyday city errands.

Negated products. Heavy-duty insulated winter boots.
Target products. Lightweight everyday walking shoes.

Type 3 Example: Negative Product Refinement

User query. I have a terrible sweet tooth and usually binge on rich chocolate wafer cookies or heavy chocolate-chunk protein bars at my desk. I want to replace those with simpler, natural snacks like honey-crystallized tropical fruits or just some simple dry-roasted nuts with herbs.

Negated products. Chocolate cookies; protein bars.
Target products. Dried fruits; dry-roasted nuts.

D.2.2. MULTI-AD QUERIES

Multi-ad queries are designed to simulate long-form or multi-need responses requiring diverse recommendations across multiple categories. Each query contains several distinct product needs, and the system is evaluated on whether its selected advertisements cover the requested categories.

Multi-Ad Example

User query. I've recently started intermittent fasting and often feel hungry in the afternoon, but I don't want anything too greasy. Can you recommend a healthy snack option to tide me over? Also, I get really drowsy when working from home in the evening and need something to wake me up that isn't as intense as an energy drink. Finally, my neck has been sore lately, and I'd like to upgrade my desk lamp to something better for eye protection.

Target categories. Snacks; beverages; home & living.

Multi-Ad Example

User query. I'm heading to the beach for a three-day camping trip next week. I need a lightweight outer layer that can keep me warm at night but won't take up much luggage space. Since we'll be walking a lot on sand and rocks, can you suggest a pair of sandals that drain water well? And with all that sun exposure, I'd like a high-protection facial sunscreen that won't leave a white cast and is easy to reapply.

Target categories. Tops; footwear; skincare.

The multi-ad setting is useful because a method can fail even when each individual need is simple. In particular, retrieval-based baselines may collapse onto the most salient category or repeatedly select near-duplicate ads, while LLM-only methods may overlook less prominent needs in a long query.

Appendix E. Prompt Templates

This section reports the prompt templates used across the different stages of the proposed framework. Variables such as the user query x, the current response context, and advertiser descriptions d_i are dynamically inserted during inference. We present the prompts in a structured format to make the role of each instruction clear. The implementation uses semantically equivalent templates, with minor wording changes depending on whether the single-turn or dynamic multi-turn setting is used.

E.1. Stage 1: Keyword Generation Prompt

The first-stage prompt extracts a small set of product-category keywords from the user query. These keywords are used only for coarse candidate filtering; the final ad selection is performed in the second stage.

Stage 1: Keyword Generation

System instruction. Extract exactly two to three relevant product-category keywords for ad matching.

Goal. The keywords will be used to select one ad for the entire upcoming assistant response. For the dynamic multi-turn setting, the scope is changed to the next one or two sentences.

Input variables.

- User query: $\{x\}$.
- Current response context: $\{context\}$, if applicable.

Instructions.

1. Extract keywords representing general product or service categories, such as `coffee_maker`, `travel_bag`, or `fitness_app`.
2. Account for the user's needs and the conversation flow.
3. If context is provided, avoid duplicating products or categories that have already been mentioned.
4. Output only comma-separated keywords, with no explanation.

Output format. `keyword_1, keyword_2, keyword_3`

E.2. Stage 2: LLM Scoring Prompt

The second-stage prompt evaluates the candidate set $\hat{\mathcal{S}}$ and asks the LLM to select the most appropriate advertiser or the no-insertion option ϕ. The probability assigned to each choice is then used to compute the LLM-based quality score.

Stage 2: LLM Scoring Prompt

System instruction. You are an expert at deciding whether and how an advertisement should be integrated into an assistant response.

Decision scope. Select one advertisement, or select no insertion, for the entire response. In the dynamic multi-turn setting, the decision scope is restricted to the next few sentences.

Input variables.

- User query: $\{x\}$.
- Current response context: $\{context\}$, if applicable.
- Candidate advertiser descriptions: $\{d_1, \ldots, d_K\}$.

Choices.

$$\begin{aligned} &\text{Choice } 0 : \text{No insertion}, \\ &\text{Choice } 1 : \{d_1\}, \\ &\text{Choice } 2 : \{d_2\}, \\ &\vdots \\ &\text{Choice } K : \{d_K\}. \end{aligned}$$

Selection criterion. Follow the user's preferences when deciding whether to insert an advertisement and which advertisement to insert. The goal is to balance user experience with the expected click-through probability of the selected ad.

Output format. `Choice [index]`

E.3. Final Response Generation Prompt

After the auction selects a winning advertiser, the final response generation prompt incorporates the winning advertiser description d_{i^*} into the generation context. The prompt encourages a smooth transition between the organic response and the sponsored content.

Final Response Generation Prompt

System instruction. You are a helpful assistant. Respond to the user's query in approximately ten sentences. If an advertisement is selected for insertion, integrate it smoothly and coherently into the response.

Input variables.

- User query: $\{x\}$.
- Current response context: $\{context\}$, if applicable.
- Selected advertiser: $\{ad_name\}$.
- Advertiser description: $\{d_{i^*}\}$.

Dynamic chat-mode instruction. Continue the response within three sentences. In these sentences, naturally incorporate $\{ad_name\}$ without mentioning other brands. Maintain local coherence with the preceding context and improve the overall flow of the response. The advertiser description is: $\{d_{i^*}\}$.

The prompt templates are intentionally simple and do not require task-specific finetuning. This design choice makes the mechanism easier to adapt to changing advertiser inventories, since new advertisers can be added by updating the candidate descriptions rather than retraining the model.

Appendix F. Additional Case Studies

This section provides qualitative examples that complement the aggregate experimental results in the main paper. The goal is to illustrate the types of query understanding failures that arise in LLM-mediated ad selection, especially when user intent is expressed through implicit constraints, intent shifts, or negated product mentions. For each case, we report the user query, the ground-truth advertisement or target categories, the predictions made by each method, and a brief analysis of the observed failure modes.

F.1. Single-Ad Queries

Single-ad queries require the mechanism to identify one most appropriate advertiser from the candidate inventory. The following examples highlight cases where surface-level semantic similarity is insufficient, because the correct ad depends on understanding constraints, negation, or the user's final intent after an initial contextual description.

F.1.1. CASE A: COMPLEX INTENT ANALYSIS

Case A: Complex Intent Analysis

User query. My gums are a bit sensitive today so I can't eat anything with a hard, brittle crunch. I'm craving a rich, aged dairy flavor in a format that just melts on the tongue without much chewing.

Ground truth. VelvetPuff Cheese Corn (Snacks).

Analysis. This case illustrates why surface-level semantic similarity is insufficient for ad selection. The query contains several partial signals, such as softness and dairy flavor, but the relevant commercial intent is a light snack that requires little chewing. The combined method preserves this intent, whereas embedding-based baselines are distracted by isolated semantic matches.

Table 6: Predictions for Case A.

Method	Prediction	Interpretation
LERA	**VelvetPuff Cheese Corn**	Correctly identifies the need for a soft, melt-in-mouth cheese-flavored snack.
LERA (LLM-Only)	**VelvetPuff Cheese Corn**	Also captures the user's nuanced constraint.
LERA (Cosine)	FlexFuel Chocolate Protein Shake	Misinterprets the query as seeking a drink rather than a snack.
LERA (Keyword-Match)	AegeanCream Strained Greek Yogurt	Overemphasizes dairy-related semantics while missing the snack context.

F.1.2. CASE B: NATURAL INTENT SHIFT

Case B: Natural Intent Shift

User query. My daily commute is brutal, but at least my Thinsulate-lined winter boots keep my feet warm and give me traction in the deep snow. My upper body is suffering, though. A regular lightweight jacket won't cut it; I need a heavy-duty, wind-blocking down coat that is specifically rated for sub-zero temperatures.

Ground truth. ArcticShield Parka (Tops).

Table 7: Predictions for Case B.

Method	Prediction	Interpretation
LERA	**ArcticShield Parka**	Correctly identifies the need for a winter coat despite the initial mention of boots.
LERA (LLM-Only)	GoldenBean Espresso Machine	Fails to identify the relevant apparel need.
LERA (Cosine)	FrostLine Insulated Winter Boots	Over-weights the initially mentioned footwear context.
LERA (Keyword-Match)	FrostLine Insulated Winter Boots	Also fails to recognize the intent shift from footwear to outerwear.

Analysis. This case highlights the challenge of natural intent shift. The user mentions winter boots to establish context, but the true need is for a sub-zero winter coat. Embedding-based methods are trapped by the earlier product mention, while the combined method correctly focuses on the user's final request.

F.1.3. CASE C: NATURAL INTENT SHIFT

Case C: Natural Intent Shift

User query. For my daily nutrition at home, I rely heavily on vacuum-sealed sous-vide salmon and collagen-rich bone broth for my protein and joint health. But when I'm commuting or heading to the gym, I don't have time to heat those up. I need a shelf-stable, high-protein snack, like a chewy oat bar with whey or ultra-crispy spiced legumes.

Ground truth. NutRidge Protein Granola Bars (Snacks).

Table 8: Predictions for Case C.

Method	Prediction	Interpretation
LERA	**NutRidge Protein Granola Bars**	Correctly recognizes the need for a portable high-protein snack.
LERA (LLM-Only)	OceanPrime Seared Salmon	Over-focuses on the user's existing at-home meal routine.
LERA (Cosine)	OceanPrime Seared Salmon	Is dominated by the prominent salmon mention in the query.
LERA (Keyword-Match)	OceanPrime Seared Salmon	Fails to shift from home meals to on-the-go snacks.

Analysis. The query first describes products the user already consumes, but those products are unsuitable for the new commuting and gym context. All baseline methods are misled by the salient food-related product mentions, whereas the combined method identifies the true requirement: a shelf-stable and portable high-protein snack.

F.1.4. CASE D: NEGATIVE PRODUCT REFINEMENT

Case D: Negative Product Refinement

User query. I've been trying to diet with those plain brown rice cakes, but they are just too airy and neutral. I need to stop eating diet food that tastes like air. I'm craving a really robust, slow-baked crunch that can hold up to some heavy dips.

Ground truth. AmberGrain Pretzels (Snacks).

Analysis. This example demonstrates the importance of modeling negation. The user explicitly mentions rice cakes, but only to reject them. Cosine similarity treats this mention as positive relevance evidence, while the combined method correctly interprets the contrast between the rejected product and the desired product attributes.

F.2. Multi-Ad Queries

Multi-ad queries evaluate whether a method can cover multiple distinct user needs within a single long-form response. We report category coverage, defined as the fraction of requested product categories represented by the selected advertisements. This metric captures whether the mechanism diversifies ad selection across the user's stated needs, rather than repeatedly selecting ads from a

Table 9: Predictions for Case D.

Method	Prediction	Interpretation
LERA	**AmberGrain Pretzels**	Correctly recognizes the desire to move away from airy diet food toward a substantial snack.
LERA (LLM-Only)	**AmberGrain Pretzels**	Also captures the negative sentiment toward rice cakes.
LERA (Cosine)	CrispRice Brown Rice Cakes	Selects the product type that the user explicitly rejects.
LERA (Keyword-Match)	CrispFalls Potato Chips	Captures the general snack category but misses the need for a sturdy, dip-friendly pretzel.

single dominant category. It should be interpreted as a category-level metric rather than an exact product-level satisfaction metric.

F.2.1. CASE E: REMOTE WORK CARE PACKAGE

Case E: Remote Work Care Package

User query. I'm putting together a care package for my sister who just started a new remote job. She loves crunchy snacks with bold flavors. She also mentioned her apartment gets stuffy, so a small quiet desk fan would help. And she's really into herbal teas lately. Any recommendations across those three categories?

Expected categories. Snacks; Beverages; Home & Living.

Table 10: Category coverage for Case E.

Method	Coverage	Selected Categories	Main Failure Mode
LERA	**100%**	Snacks, Beverages, Home & Living	Covers all requested categories.
LERA (LLM-Only)	33%	Home & Living, Home & Living, Home & Living	Collapses entirely to home appliances; covers one requested category (Home & Living) but misses snacks and beverages.
LERA (Cosine)	67%	Home & Living, Beverages, Beverages	Misses the requested snack category.
LERA (Keyword-Match)	67%	Beverages, Snacks, Food & Dining	Misses the requested home appliance category.

Output Comparison. Below we present the actual selected advertisements for each method in Case E.

Case E: Output by Method — LERA

Selected ads.

1. LavaThin Potato Chips (Snacks)
2. SparkLeaf Botanical Tea (Beverages)
3. EverCool Desk Fan (Home & Living)

Model response. For crunchy snacks with bold flavors, I'd recommend trying LavaThin Potato Chips—these paper-thin, kettle-cooked chips pack an intense chili punch that builds with each bite, perfect for someone who loves heat with a smoky habanero finish. They're bold enough to wake up the taste buds during a mid-afternoon work slump. To complement those vibrant flavors, include a few cans of SparkLeaf Botanical Tea—its light effervescence and artfully blended herbs like butterfly pea flower and lemongrass offer a refreshing, caffeine-free pause that feels both calming and luxurious, ideal for unwinding between video calls. Since she's working in a stuffy apartment, a compact fan that delivers steady, quiet airflow without disrupting focus is essential—something like the EverCool Desk Fan, which gently oscillates and offers adjustable speeds to keep her cool and comfortable all day.

Case E: Output by Method — LERA (LLM-Only)

Selected ads.

1. EverCool Desk Fan (Home & Living)
2. NimbusLight Desk Lamp (Home & Living)
3. PureCycle HEPA Air Purifier (Home & Living)

Model response. For a snack, try wasabi peas or spicy sriracha roasted chickpeas—they're crunchy, bold, and perfect for mid-afternoon energy. A soothing herbal tea blend like chamomile-lavender or peppermint-ginger will help her unwind after long screen sessions. To keep her workspace fresh and comfortable, a compact fan paired with the NimbusLight Desk Lamp can make a real difference—its adjustable, glare-free lighting helps reduce eye strain during long workdays, while the customizable color temperature supports focus and relaxation as needed. Since her apartment can feel stuffy, adding a PureCycle HEPA Air Purifier will quietly refresh the air with its true HEPA and activated carbon filters, removing pollen and lingering odors without disrupting her focus.

Case E: Output by Method — LERA (Cosine)

Selected ads.

1. SwiftBlend Portable Blender (Home & Living)
2. SparkLeaf Botanical Tea (Beverages)
3. FunkBrew Ginger Kombucha (Beverages)

Model response. For her herbal tea enjoyment, consider including a selection of caffeine-free blends like chamomile, peppermint, and rooibos—perfect for staying relaxed and focused during work hours. Since she's health-conscious and often on the move, the SwiftBlend Portable Blender would be a fantastic addition, letting her whip up refreshing iced tea infusions in seconds. Elevate her tea experience with SparkLeaf Botanical Tea—its lightly carbonated, unsweetened blends with exotic botanicals offer a crisp, sophisticated sip that's both refreshing and naturally energizing.

Case E: Output by Method — LERA (Keyword-Match)

Selected ads.

1. SparkLeaf Botanical Tea (Beverages)
2. GlazePeak Dried Fruits (Snacks)
3. DawnOats Instant Oatmeal Cups (Food & Dining)

Model response. For herbal teas, I'd recommend SparkLeaf Botanical Tea—its lightly carbonated, unsweetened blends with exotic botanicals offer a refreshing, caffeine-free pick-me-up perfect for long workdays. When she's craving something sweet and satisfying between meetings, GlazePeak Dried Fruits are a perfect treat—each bite delivers a burst of tropical flavor with a honey-kissed chewiness that feels like an indulgent snack but fuels her like a smart choice. And for cozy mornings when she wants warmth without the mess, a cup of DawnOats Instant Oatmeal offers a nourishing, no-fuss breakfast.

Analysis. This case demonstrates the importance of balancing category coverage in multi-need queries. While LLM-Only completely collapses to home appliances (selecting desk fan, desk lamp, and air purifier), and Cosine misses snacks by selecting a blender instead, the combined method successfully identifies one relevant product from each requested category: spicy potato chips for bold snacking, botanical tea for herbal tea enthusiasts, and a quiet desk fan for the stuffy apartment. This shows how the retrieval-and-scoring framework prevents over-concentration in semantically dominant categories.

F.2.2. CASE F: BEACH VACATION ESSENTIALS

Case F: Beach Vacation Essentials

User query. For my upcoming beach vacation, I want a lightweight linen shirt that breathes well in humidity. I also need a solid SPF moisturizer for my face that won't feel greasy. And a pair of comfortable sandals for walking on the boardwalk would round out my packing list. Suggestions?

Expected categories. Tops; Skincare; Footwear.

Output Comparison. Below we present the actual selected advertisements for each method in Case F.

Case F: Output by Method — LERA

Selected ads.

1. BreezeLinen Summer Shirt (Tops)
2. SunVeil SPF 50 Face Moisturizer (Skincare)
3. BerryTint SPF Lip Balm (Skincare)

Model response. The BreezeLinen Summer Shirt is an ideal choice for your beach getaway—crafted from lightweight, natural linen that's designed to breathe and wick moisture, keeping you cool even in high humidity. Its relaxed yet refined look transitions effortlessly from seaside strolls to dinner on the boardwalk. To complete your daytime routine, pair it with SunVeil SPF 50 Face Moisturizer—a silky, non-greasy formula that blends broad-spectrum mineral protection with daily hydration, leaving no white cast. For your walks along the shore, consider a pair of contoured, cushioned sandals with arch

Table 11: Category coverage for Case F.

Method	Coverage	Selected Categories	Main Failure Mode
LERA	67%	Tops, Skincare, Skincare	Captures apparel and sun protection but misses footwear.
LERA (LLM-Only)	**100%**	Tops, Skincare, Footwear	Covers all requested categories.
LERA (Cosine)	33%	Tops, Tops, Tops	Collapses entirely to apparel; covers one requested category (Tops) but misses skincare and footwear.
LERA (Keyword-Match)	33%	Skincare, Skincare, Skincare	Collapses entirely to skincare; covers one requested category (Skincare) but misses apparel and footwear.

support—perfect for long stretches on sunbaked boardwalks. Don't forget to protect your lips too—a swipe of BerryTint SPF Lip Balm delivers a sheer berry tint, SPF 15, and nourishing shea.

Case F: Output by Method — LERA (LLM-Only)

Selected ads.

1. BreezeLinen Summer Shirt (Tops)
2. SunVeil SPF 50 Face Moisturizer (Skincare)
3. RiverGrip Hiking Sandals (Footwear)

Model response. For a lightweight linen shirt that handles humidity beautifully, consider the BreezeLinen Summer Shirt, designed with a relaxed yet refined fit that transitions effortlessly from beach strolls to dinner by the shore. Its moisture-wicking fabric stays cool against the skin, offering superior breathability without sacrificing style. Pair it with a matte SPF like SunVeil SPF 50 Face Moisturizer—a featherlight daily formula that blends broad-spectrum mineral protection with hydration. When it comes to sandals, look for a pair that combines comfort with serious traction. The RiverGrip Hiking Sandals feature adjustable webbing straps and a contoured footbed designed to support all-day walking, whether you're on sun-baked boardwalks or coastal trails.

Case F: Output by Method — LERA (Cosine)

Selected ads.

1. BreezeLinen Summer Shirt (Tops)
2. BreezeVest Packable Down Vest (Tops)
3. CottonCrest Crewneck T-Shirt (Tops)

Model response. The BreezeLinen Summer Shirt is an ideal choice for your beach getaway—its lightweight, open-weave fabric allows maximum airflow, keeping you cool and comfortable even in high humidity. For cooler evenings by the shore, consider packing the BreezeVest Packable Down Vest—it delivers cozy, lightweight warmth without bulk and conveniently compresses into its own pocket. When you're looking for a reliable, go-to layer under your linen shirt or on low-key beach days, nothing beats the clean simplicity of a well-made crewneck—like the ones from CottonCrest that use heavyweight cotton for a structured feel that molds to you over time.

Case F: Output by Method — LERA (Keyword-Match)

Selected ads.

1. SilkBody Shea Body Lotion (Skincare)
2. SunVeil SPF 50 Face Moisturizer (Skincare)
3. BerryTint SPF Lip Balm (Skincare)

Model response. For your beach days and sun-kissed skin, don't forget to care for your body with the same thoughtfulness—SilkBody Shea Body Lotion deeply nourishes with shea butter and ceramides, absorbing quickly without leaving a heavy residue. To keep your face protected and feeling fresh, try SunVeil SPF 50 Face Moisturizer—a lightweight daily formula that combines broad-spectrum mineral protection with hydration. When you want subtle color and sun defense in one easy step, reach for a tinted lip balm that glides on smoothly, shielding your lips with SPF 15 while leaving a natural berry hint.

Analysis. This case reveals dramatic failures in embedding-based methods. Cosine collapses entirely to apparel (selecting a linen shirt, down vest, and cotton T-shirt), while Keyword-Match collapses entirely to skincare (selecting body lotion, face moisturizer, and lip balm). Both completely miss the user's footwear need for boardwalk sandals. LLM-Only achieves perfect coverage by selecting appropriate products from all three categories. The combined method achieves 67% coverage, capturing the shirt and face moisturizer but missing footwear—it instead selects a lip balm as the third item, suggesting that small, low-cost sun protection items can compete with footwear in the auction mechanism when they share similar semantic features.

F.2.3. CASE G: WORKOUT ROUTINE ESSENTIALS

Case G: Workout Routine Essentials

User query. I'm starting a new workout routine and need a good pair of cross-training shoes for the gym. I also want a plant-based protein shake that doesn't taste artificial. And a sweat-wicking performance tee that doesn't cling awkwardly. Any recommendations?

Expected categories. Footwear; Beverages; Tops.

Output Comparison. Below we present the actual selected advertisements for each method in Case G.

Table 12: Category coverage for Case G.

Method	Coverage	Selected Categories	Main Failure Mode
LERA	**100%**	Footwear, Tops, Beverages	Covers all requested categories.
LERA (LLM-Only)	**100%**	Footwear, Tops, Beverages	Covers all requested categories.
LERA (Cosine)	33%	Beverages, Snacks, Snacks	Collapses to nutrition products; covers one requested category (Beverages) but misses footwear and apparel entirely.
LERA (Keyword-Match)	67%	Footwear, Footwear, Beverages	Duplicates footwear; misses the requested apparel category.

Case G: Output by Method — LERA

Selected ads.

1. StableGrip Cross-Trainers (Footwear)
2. AeroTech Performance Tee (Tops)
3. FlexFuel Chocolate Protein Shake (Beverages)

Model response. For your cross-training needs, the StableGrip Cross-Trainers offer the perfect blend of support and stability, with a flat sole that ensures solid footing during lifts and lateral movements. Their durable design enhances balance and control, making them ideal for everything from squats to agility drills in the gym. With a focus on function and performance, pair your new shoes with the AeroTech Performance Tee—its seamless, synthetic build and laser-cut ventilation keep you cool and dry, moving naturally with your body during intense workouts without the cling or weight of cotton. After pushing hard in your session, refuel with a FlexFuel Chocolate Protein Shake—its rich, creamy texture and 30g of plant-powered protein deliver satisfying recovery nutrition without the chalky aftertaste or artificial flavors you're trying to avoid.

Case G: Output by Method — LERA (LLM-Only)

Selected ads.

1. FlexFuel Chocolate Protein Shake (Beverages)
2. StableGrip Cross-Trainers (Footwear)
3. AeroTech Performance Tee (Tops)

Model response. For a plant-based protein shake that actually satisfies, try the FlexFuel Chocolate Protein Shake—its rich, creamy texture and 30g of protein make it a standout choice for post-workout recovery without any artificial aftertaste. Pair your shake with the StableGrip Cross-Trainers, engineered with a flat, stable sole to support heavy lifts and dynamic lateral movements, so you stay grounded during every rep, set, or circuit. Complete your gear with a performance tee that keeps pace with your

intensity—like the AeroTech Performance Tee, crafted with seamless construction and laser-cut ventilation for maximum airflow and zero cling.

Case G: Output by Method — LERA (Cosine)

Selected ads.

1. FlexFuel Chocolate Protein Shake (Beverages)
2. NutRidge Protein Granola Bars (Snacks)
3. GlazePeak Dried Fruits (Snacks)

Model response. For a plant-based protein shake that tastes rich and satisfying without any artificial aftertaste, try the FlexFuel Chocolate Protein Shake—its creamy texture and 30g of protein make it perfect for recovery after a tough gym session. When you're heading to the gym or need a quick boost between sets, pack a NutRidge Protein Granola Bar—each chewy bite combines whey isolate, hearty oats, and rich dark chocolate chunks for a satisfying snack. After your workout, refuel with a handful of GlazePeak Dried Fruits—sun-ripened tropical slices kissed with natural honey for a burst of energizing sweetness.

Case G: Output by Method — LERA (Keyword-Match)

Selected ads.

1. StableGrip Cross-Trainers (Footwear)
2. FlexFuel Chocolate Protein Shake (Beverages)
3. PeakFlow Running Shoes (Footwear)

Model response. For your cross-training needs, I highly recommend the StableGrip Cross-Trainers—they're designed with a flat, stable sole that provides excellent support during weightlifting and lateral movements, making them ideal for the varied demands of a gym workout. After pushing hard, refuel with a FlexFuel Chocolate Protein Shake—its rich, creamy texture and 30g of plant-powered protein deliver satisfying recovery nutrition without any chalky aftertaste. If your routine ever shifts toward longer runs or speed training, consider stepping into a pair of PeakFlow Running Shoes—lightweight, race-ready footwear with responsive cushioning and carbon-fiber plates for efficient energy return.

Analysis. This case showcases the most dramatic failure mode of embedding-based methods: Cosine completely misses both footwear and apparel, selecting only nutrition products (protein shake, granola bars, and dried fruits). Keyword-Match duplicates footwear categories by selecting cross-trainers and running shoes, failing to capture the sweat-wicking performance tee. Both the combined method and LLM-Only achieve perfect coverage, selecting appropriate cross-training shoes, performance tee, and protein shake. This illustrates that when user needs span distinct functional categories (footwear, apparel, nutrition), embedding similarity can fail to maintain proper category balance.

Across these multi-ad examples, a consistent pattern emerges: embedding-based baselines tend to collapse onto a single semantically dominant category, whether it's matching surface-level keywords or clustering in embedding space. The combined method's two-stage approach—retrieval followed by LLM-based scoring—provides a more robust mechanism for distributing selections

across the user's expressed needs, even when those needs span fundamentally different product categories.

Appendix G. Omitted Proof of Proposition 1

Restatement of Proposition 1. *Assume that the relevance scores used by the allocation rule are independent of advertiser i's reported bid except through the bid-weighted ranking rule. Further assume that $o_i(y)$ and $\hat{v}_i(y)$ are independent conditioned on the ad being displayed, and that the prior estimate is unbiased, i.e., $v_i = \mathbb{E}_y[\hat{v}_i(y)]$. Then the two-stage auction mechanism with the payment rule above is incentive compatible for a utility-maximizing advertiser.*

Proof Under the independence assumption and unbiased prior estimates, the expected utility of advertiser i can be written as

$$\begin{aligned} u_i &= \mathbb{E}_y\left[(\hat{v}_i(y) - p_i) \cdot o_i(y)\right] \\ &= \mathbb{E}_y\left[\hat{v}_i(y) - p_i\right] \cdot \mathbb{E}_y\left[o_i(y)\right] \\ &= (v_i - p_i) \cdot \bar{o}_i, \end{aligned}$$

where $\bar{o}_i = \mathbb{E}_y[o_i(y)]$ denotes the expected click probability.

Because the allocation rule is monotone in advertiser i's bid, there exists a critical payment p_i^{crit} such that advertiser i wins if and only if $b_i \geq p_i^{\text{crit}}$. This critical value is determined by the bids and scores of other advertisers, together with the organic scores used by the mechanism, and is independent of advertiser i's reported bid conditional on winning.

We consider two cases.

Case 1: $v_i \geq p_i^{\text{crit}}$. If advertiser i bids truthfully, namely $b_i = v_i$, then the advertiser wins because $b_i \geq p_i^{\text{crit}}$. The resulting utility is

$$(v_i - p_i^{\text{crit}})\bar{o}_i \geq 0.$$

Any deviation $b_i' \geq p_i^{\text{crit}}$ leads to the same allocation and the same payment, and therefore yields the same utility. Any deviation $b_i' < p_i^{\text{crit}}$ causes the advertiser to lose, resulting in utility 0, which is weakly worse than truthful bidding.

Case 2: $v_i < p_i^{\text{crit}}$. If advertiser i bids truthfully, then $b_i = v_i < p_i^{\text{crit}}$, so the advertiser loses and obtains utility 0. Any deviation $b_i' < p_i^{\text{crit}}$ leads to the same outcome. Any deviation $b_i' \geq p_i^{\text{crit}}$ causes the advertiser to win, but the payment is p_i^{crit}. Since $p_i^{\text{crit}} > v_i$, the resulting utility is

$$(v_i - p_i^{\text{crit}})\bar{o}_i < 0,$$

which is strictly worse than losing.

Therefore, bidding below the true value can only forgo a nonnegative utility opportunity, while bidding above the true value can only create the possibility of winning at a price exceeding the advertiser's value. Truthful bidding is thus a dominant strategy in expectation. ■

Remark 2 (On the independence assumption) *The independence assumption in Proposition 1 is introduced as a simplifying condition for the theoretical analysis rather than as a literal claim about all real-world ad markets. In our setting, the randomness arises from the LLM generation process: even for a fixed query and a fixed winning advertiser, the final response y may vary because of*

stochastic decoding. Both the click-through outcome $o_i(y)$ and the ex-post advertiser value $\hat{v}_i(y)$ are therefore functions of the realized response.

When the product price is fixed, $\hat{v}_i(y)$ can be interpreted as post-click conversion value, so the assumption amounts to separating CTR and post-click value conditional on display. We view this as a reasonable first-order approximation because the relationship between these two quantities can be highly context-dependent: some responses may increase clicks but reduce downstream conversion, for example by using overly aggressive or misleading phrasing, whereas others may increase both click propensity and conversion intent by providing clear and well-matched recommendations. This relationship can further depend on product characteristics and user context.

Hence, rather than assuming a universal positive correlation, we adopt independence as a tractable benchmark under which the critical-value argument is transparent. Relaxing this assumption and analyzing richer dependence structures between click probability and advertiser value is an important direction for future work.

Appendix H. Limitations and Broader Impacts

Limitations. This paper studies auction design within the retrieve-then-generate paradigm. This choice reflects a practical deployment constraint: the platform can keep the expensive full-response generation step mostly unchanged while improving the auction signal used before generation. However, it also means that our mechanism does not directly optimize the full generated response jointly with advertiser allocation. Future work could explore tighter integration between generation and mechanism design while preserving tractable payment rules.

Due to the lack of an off-the-shelf open-source benchmark for LLM-mediated ad auctions, our experiments rely on a synthetic advertiser-query dataset constructed with LLM assistance and organized into controlled evaluation groups. This design allows us to test specific failure modes such as negation, intent shifts, and category collapse, but it cannot fully capture the distributional complexity of real advertising systems. Larger-scale evaluation with commercial inventories, historical click data, and human judgments of response quality would provide a stronger test of deployment readiness.

The theoretical result is also based on a simplifying independence assumption between click outcome and ex-post advertiser value, as discussed in Appendix G. We view this assumption as a tractable benchmark that makes the critical-value structure transparent, rather than as a complete behavioral model of all ad markets. Relaxing this assumption and incorporating richer dependence between response wording, click probability, conversion value, and advertiser utility is an important direction for future work.

Broader impacts. LERA is designed to improve the commercial effectiveness of LLM-based chatbots by making ad insertion more context-aware and less repetitive. Used responsibly, such mechanisms could help platforms monetize generative services while selecting advertisements that better match user intent and preserve response coherence. At the same time, chatbot advertising may be perceived as intrusive, especially when sponsored content is not clearly disclosed or when the system over-prioritizes advertiser value over user benefit. Practical deployments should therefore include transparency requirements, user controls, frequency limits, and safeguards against misleading or manipulative ad integration.