Online Advertisements with LLMs: Opportunities and Challenges

SOHEIL FEIZI, MOHAMMADTAGHI HAJIAGHAYI, KEIVAN REZAEI, SUHO SHIN
University of Maryland

In modern online platforms, advertisement plays a crucial role in subsidizing their operational costs. With the emerging advances in generative AI, this position piece explores the potential of running online advertisement systems for Large Language Models (LLMs). Our position is that, the existing framework for modern search advertisement system does not carry over to LLM advertisement. To understand how one could run advertisement systems for LLMs by integrating ads into their outputs and facilitate future research in this area, we introduce a generic framework for LLM advertisement. We explore several aspects of design choices within our framework and propose potential research questions to operate a practical system.

Categories and Subject Descriptors: J.4 [**Social and Behavior Sciences**]: Economics

General Terms: Algorithms, Design, Economics, Theory

Additional Key Words and Phrases: Online advertising, auction design, large language model

1. INTRODUCTION

In the vast landscape of online search engines, the role of advertisements has become pivotal, shaping the digital experience for users globally. The enormity of the market, with trillions of dollars at play, shows the economic significance of advertising, *e.g.,* the market size of search advertisement as of 2020 was valued at USD 164.12 billion [Insight 2022]. Online advertising not only brings revenue to platform companies but also plays a crucial role in subsidizing free access to information and services for every individual. The democratization of education, information, and other resources is made possible through ad-driven models, enabling individuals to not only consume content for free but also contribute to the digital ecosystem. Additionally, the symbiotic relationship between advertisements and content creation creates a feedback loop, fostering economic growth. As a side note, even subscription-based streaming platforms like Netflix are starting ad-supported plans [Netflix 2022].

On the other hand, recently, large language models (LLMs) [Brown et al. 2020; Anil et al. 2023; Thoppilan et al. 2022] have gained widespread adoption among users, serving various functions including question answering, content generation, translation, code completion, and more [Nijkamp et al. 2022; Fried et al. 2022; Wang

Authors' addresses: (sfeizi,hajiagha,krezaei,suhoshin)@umd.edu
Version history. Nov 2023: first version. Feb/Apr/Sep 2024: added discussions on recent works. This version: revised presentation. Full version is available at [Feizi et al. 2023].
We thank Sébastien Lahaie, Irene Lo, and Sam Taggart for valuable feedback. The authors are partially supported by DARPA QuICC, NSF AF:Small #2218678, NSF AF:Small #2114269, Army-Research Laboratory (ARL) #W911NF2410052, and MURI on Algorithms, Learning and Game Theory.

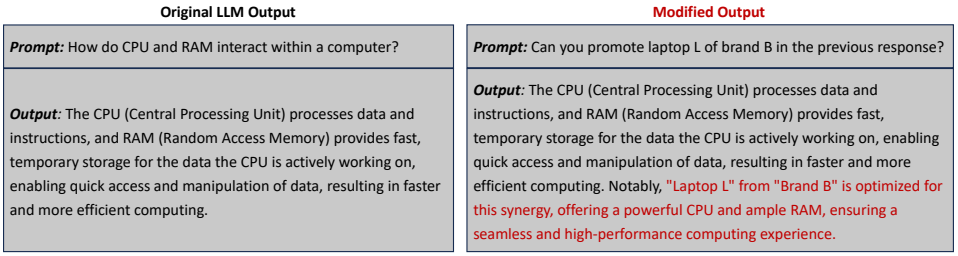

Fig. 1: An example of providing **unstructured** advertisement in LLM output. The left figure shows the original prompt to and output from ChatGPT 4. The right figure shows the prompt used to **incorporate an ad**.

and Komatsuzaki 2021; Liu et al. 2023]. The proliferation of AI-driven assistant language models, such as ChatGPT, has contributed to a growing trend wherein individuals increasingly use such models to address their inquiries, occasionally replacing traditional search engines as their primary information-seeking tool. According to [PCMag 2023], even for now, 35% of casual users say they find LLMs to be more helpful in finding information than search engines. It is obvious that such a trend will be accelerating in the near future as well. The substantial usage volumes stemming from diverse users would induce companies offering these tools, which we call *LLM providers*, to contemplate revenue generation through advertising [AdWeek 2023; Crunch 2023; Microsoft 2023]. Consequently, an interesting and fundamental question arises:

How can **LLM providers** make revenue by running an online **advertisement** on their services?

The concept of online advertising has been extensively studied within the realm of search engines, where auctions are conducted among advertisements from advertisers when a user inputs a query. This paper focuses on the prospect of transposing this online advertising model and auction framework to the context of large language models. We further discuss technical challenges and potential framework to run online advertisement system in LLM, thereby calling academic and industrial researchers to the area of importance.

Search advertising. To better explain fundamental differences between standard search advertising (SA)[1] and LLM advertising (LLMA), we briefly introduce how standard SA works [Lahaie et al. 2007]. (1) *Bidding*: In SA, the owner of each ad i writes bid $b_i \in \mathbb{R}_{\geq 0}$ on targeting *keyword* for $i \in [n]$, which can be a set of keywords. (2) *Output generation*: The platform first decides *how many slots* to allocate for ads in the search engine results page (SERP), say k. (3) *Prediction*: Given k slots in SERP, the platform then predicts the click-through-rate (CTR) α_{ij} when ad i is

[1]Its mechanism design problem is often called sponsored search auction (SSA).

allocated in slot j. (4) *Auction*: The platform then optimizes

$$\max_{x\in[0,1]^{n\times k}} \sum_{i=1}^{n}\sum_{j=1}^{k} \alpha_{ij} b_i x_{ij}, \tag{1}$$

where $x = (x_{ij})_{i\in[n],j\in[k]}$ is the (possibly randomized) allocation vector such that $x_{ij} = 1$ if ad i is allocated in slot j given the constraint $\sum_{i=1}^{n} x_{ij} \leq 1$ for every $j \in [k], i \in [n]$. The platform then charges each ad according to some pre-defined payment rule.

Overall, whenever a user arrives in the platform and searches a keyword, the set of ads related to the keyword are determined. Then, the platform collects corresponding bids of the selected ads, decides the number of slots k, predicts CTR, and the auction runs.

Motivating example. How would LLMA be fundamentally different from SA? We start with illustrative scenarios where a user asks a technical question about computers (Figure 1). Without advertisement, an LLM would typically generate a response to address the user's query. To incorporate advertisements in the generated output, there is a spectrum of possibilities for including ad content, such as: (a) putting the ads outside the response but visibly in the user interface, (b) incorporating the ads within the generated output directly. The ads in option (a) can be treated as display ads, and may be relatively easy to handle using some of the vast amount of prior work on display ads. However, (b) is more similar to a sponsored search ad.[2] We will focus on approach (b), which will entail fundamental challenges that have not arisen in traditional SA. [3]

SA versus LLMA. Recall the process of bidding, output generation, prediction, and auction in the SA mentioned before, and imagine implementing those modules for LLMA.

For the bidding module, since the LLM's query could be far much complicated than just a single keyword in SA, how could the advertisers express their willingness-to-pay for each query? Essentially, each advertiser might want to significantly adjust its bid based on how much they find the query to be relevant to its ad. Further, given that the marketing impact will significantly depend on how the LLM incorporates the ad in the output, it is not even clear what is the advertiser's *value* for being included in the ad. If the advertiser's value depends on the generated output, how could the advertiser reflect their *willingness-to-pay* with respect to the output, which might not be accessible in advance? Even further, how can we generate output that smoothly incorporates the ad without hurting the user experience while satisfying the advertiser?

For the prediction module, most SA run an online learning algorithm to update the ad's feature vector with respect to user context [McMahan et al. 2013]. This was

[2]Search ads usually capture user attention better than display ads, *e.g.,* almost 50% more views [Outbrain 2023].

[3]If ads are included within the generated output, there are possibilities of structured outputs (e.g., ads replacing one of the elements in a given list of elements), or the ads can be included beyond the output (similar to display ads). We focus on unstructured output, as it is broadly applicable, while structured output may be addressed using the standard SA framework.

possible because the ad images, hyperlinks, and more generally how they appear in the SERP remain the same across many user interactions. LLM, however, could incorporate ads in a very different manner for each query, which makes it difficult for the LLMA to *learn the CTR*. Also, since the ads are merged into the generated output, they significantly affect user experience. How, then, can we guarantee and measure user satisfaction?

Finally, which kind of auction format should the LLMA run? How can the LLMA adapt for advertising multiple ads in a single output? What would be a reasonable analogue of the autobidding system prevalent in the modern online ad system?

All these questions are not straightforward to answer and to our knowledge have not been formally discussed in the literature.

Outline. Given the outlined uniqueness and differences with SA, we expect that LLMA requires a number of research questions to operate in practice. To understand LLMA and present its technical challenges compared to the SA, we first introduce a generic framework to operate LLMA.[4] Similar to SA, our framework consists of four modules, though the implementation of each module will be very different from SA: (i) *modification* in which the original output of LLM is modified; (ii) *bidding* that advertisers utilize to bid on the modified outputs; (iii) *prediction* in which LLMA computes required information about advertisements; and (iv) *auction* in which the advertisers compete and the final output is selected. We introduce design choices for each module, evaluated against criteria essential for a sustainable system, and discuss the research challenges inherent to each module. Our framework further enables a unified interpretation and comparison of recent approaches for LLM advertisement systems.[5]

1.1 Related works

Here, we discuss related works on LLMs, online ads, and their intersection.

Large Language Models. Advancements in AI, NLP, and conversational agents, driven by Transformer architecture [Vaswani et al. 2017], have given rise to models like GPT-3 [Brown et al. 2020] and BERT [Devlin et al. 2019]. These models revolutionize chatbots, enabling context-aware, human-like interactions across diverse domains [Abd-Alrazaq et al. 2020; Nicolescu and Tudorache 2022]. Everyday use of language models has led researchers to investigate the content generated by these models to ensure that they do not hallucinate [Guerreiro et al. 2023; Ji et al. 2023; Li et al. 2023; Zhang et al. 2023] in their outputs, and do not generate harmful or biased content [Liang et al. 2021; Navigli et al. 2023; Kirk et al. 2021; Shen et al. 2023; Weidinger et al. 2021; Liu et al. 2023]. In fact, trustworthy of LLMs is actively studied by researchers [Liu et al. 2024].

Online Advertisement. Online advertising, particularly within the context of sponsored search auctions, has evolved in recent years, with notable contributions from prior research. Sponsored search auctions have been a subject of extensive

[4]While our framework could serve as an initial step for future research, our focus is to address key questions for the practical operation of LLM advertisement

[5]We briefly discuss further perspectives on the interplay between LLMs and online advertising systems, *e.g.,* improving the user attraction by personalizing the ad images by LLMs, see Section 5 or the full paper [Feizi et al. 2023].

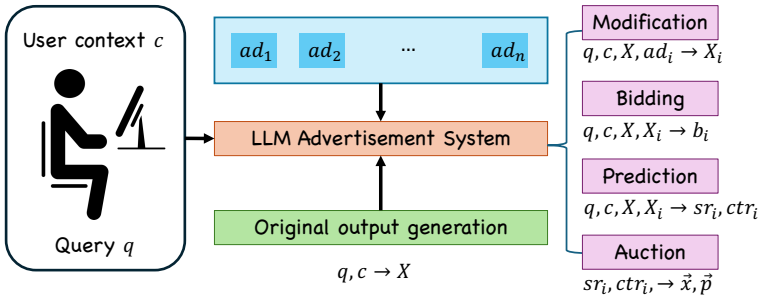

Fig. 2: Overall framework of LLMA.

investigation, emphasizing the optimization of bidding strategies and keyword relevance. [Edelman et al. 2007] provided valuable insights into the economics of sponsored search auctions, shedding light on the dynamics of keyword auctions. [Goel et al. 2009] proposed a contract auction between the advertiser and the publisher, and introduce impression-plus-click pricing for sponsored search auction as an application. We refer to the book by [Roughgarden 2010] for more details. Finally, there exists an emerging interest from the community to study ad auctions in LLMs, which we elaborate more on Section 4.5.

2. FRAMEWORK FOR LLMA

To understand how LLMA could operate in practice, given the stark differences between SA and LLMA, we here present a potential generic framework for LLMA.[6] Mainly, we focus on a scenario in which a user provides a query q to the LLM, andLet the original output by the LLM is given by X. Further, a context c captures a variety of features that are relevant to the advertisement recommendation, *e.g.,* history of the previous queries, user segment, region, and date.[7] Although the number of advertisers varies from time to time, when the user inputs the query q, we suppose that there are n advertisers (bidders) indexed by $\mathrm{adv}_1, \ldots, \mathrm{adv}_n$, each of which is equipped with a single advertisement (ad) ad_i he/she wants to post.

Overall, we divide the LLMA into 4 modules as follows based on their functionalities[8]: (i) output modification, (ii) bidding, (iii) prediction, and (iv) auction. This overall framework is illustrated in Figure 2. In short,

(1) The user writes query q possibly with context c.

(2) The LLMA generates the original output X.

(3) The **modification module** creates modified output X_i per ad.

(4) The **bidding module** generates corresponding bid b_i.

[6]While entirely different systems outside this framework may emerge, we believe it can foster future discussions to address technical challenges in LLMA.

[7]Note here that we assume LLMA could collect such information from the user similar to the standard search engines, but our model accommodates the setting without such data.

[8]These functionalities may be distributed to multiple market players such as supply side platform, display side platform, or ad exchange, as does in the current online ad eco-system.

(5) The **prediction module** predicts user satisfaction rate (SR) sr_i and click-through-rate (CTR) ctr_i.
(6) The **auction module** determines the final output and corresponding payment to charge the selected advertiser.[9]

For the rest of the section, we will explain each module and its responsibility/functionality in a sequential manner.

2.1 Modification module

The modification module generates modified output based on the ads' textual information. This module takes the pair of (q, X, c) and a set of advertisements as the input, and returns $(X_i)_{i\in[n]}$ where X_i denotes the modified output for ad_i.

Overall, we consider two design choices: [10]

(1) In the *advertiser modification* model, the role of generating the modified output is delegated to each advertiser.
(2) In the *LLMA modification* model, LLMA directly generates the modified output.

In Section 4.1, we explore ways to reduce the computational load of generating candidate outputs.

Comparison to SA. Note that the standard SA does not have a modification module explicitly as it is trivial to incorporate ads in each slot. Thus, this is a unique challenge appearing in LLMA. Further, we note that LLM is only directly used in the modification module as well as in generating the original output X, whereas other modules do not require LLM to operate them. Nevertheless, the other modules have technical challenges specific to LLMA, which we now present.

2.2 Bidding module

The bidding module generates bids based on modified outputs. It takes the query q, context c, and modified outputs $(X_i)_{i\in[n]}$ as input and outputs bids $(b_i)_{i\in[n]}$, representing each advertiser's private valuation of impressions, clicks, or conversions. We consider two design choices.

(1) In the *dynamic bidding* model, we provide the query q, context c, original output X,[11] and modified output X_i to the bidder for each query, who then returns the bid.
(2) In the *static bidding* model, each bid is based on keywords from a pre-committed contract, without further communication with the advertiser.

The static bidding model operates by extracting keywords from q, determining relevant ads, and requiring advertisers to set targeted keywords. In Section 4.2, we discuss extensions of these bidding models that allow more flexible bidding strategies. This approach can be preferred when the bid unit accurately represents the

[9] We first focus on presenting a single advertisement in the LLM output, however, generalization of the proposed framework to incorporate multiple ads at once is discussed in Section 4.4.

[10] For either case, one can generate the modified output by giving an additional query to the LLM as presented in Figure 1.

[11] An encrypted context $\hat{c}$ or no context could be considered for privacy.

advertiser's true valuation, regardless of output quality, such as clicks or conversions.

Dynamic bidding models interest advertisers with their measures to estimate the quality of modified output. For instance, if LLMA only considers CTR and ignores user experience, advertisers may wish to adjust bids based on their output quality assessment. Despite a high likelihood of clicks or conversions, poor user experience with low-quality outputs could harm satisfaction with the advertised product.

Comparison to SA. In SA, since the context of the user query is more explicitly represented as keywords, advertisers can safely bid on each relevant keyword (which SA requires). In LLMA, however, it might be difficult to extract proper keywords from the query as the query itself tends to be much longer than in traditional SA due to the flexibility in LLM's query. On the other hand, one can instruct the LLM to extract core keywords or use word embeddings to calculate the similarity between the modified output and query. Further, the generated output significantly affects the marketing impact of the ads in LLMA, whereas it is typically independent from other ads / contents shown within SERP in SA. Finally, the dynamic bidding model exhibits unique challenges of dynamically adjusting the bid with respect to the output, which could further be delegated to another market player or LLM agent.

2.3 Prediction Module

The prediction module computes the user's satisfaction rate (SR) and click-through rate (CTR). The SR measures user satisfaction with the output and influences the decision-making process of the language model to avoid disappointing outputs. The CTR represents the likelihood of the user clicking on the ad link in the output and is crucial for determining auction winners, as it directly impacts the LLMA's revenue. Specifically, if an advertiser's bidding method is cost-per-click (CPC), the expected revenue from the ad is calculated as CPC multiplied by the CTR. Overall, both SR and CTR are functions of the original output X, modified output X', query q, and context c, which returning a real value in $[0, 1]$.

Comparison to SA. Different from the traditional SA whose output is static (ad image/hyperlinks), LLMA constructs textual outputs in a complicated manner. This makes it more difficult for the prediction module to learn the CTR. Moreover, in LLMA, the prediction of user satisfaction is much more directly affected by the incorporation of the ads in the output. This is in stark contrast with SA, where user experience is usually affected only by the number of ad slots/positions in SERP. Detailed methodologies for estimating/learning these functions will be discussed in Section 4.3.

2.4 Auction module

Having computed all the required parameters, we run the auction module to determine the auction winner and the advertiser's charge. The input to the auction module is the set of tuples $(\text{bid}_i, \text{sr}_i, \text{ctr}_i)$ $i \in [n]$, representing bid amount, satisfaction rate, and click-through rate for each bidder. The auction module outputs an (possibly randomized) allocation $\vec{x} \in [0, 1]^n$ and payments $\vec{p} \in \mathbb{R}^n_{\geq 0}$. Specifically, the module determines the auction format, including the allocation function, which

selects the ad, and the payment function, calculating the advertiser's payment to LLMA.[12]

The main goal of LLMA is to maximize its long-term revenue by balancing short-term revenue with user retention. The objective is modeled as a function from bid amount, CTR, and SR to a nonnegative score for a modified output, i.e., selecting $i^* = \text{argmax}_{i \in [n]} \text{Obj}(\text{sr}_i, \text{ctr}_i, \text{bid}_i)$. We do not detail the objective function choice, given the extensive literature on sponsored search auctions. After designing the score function, an auction format should be determined, with many mechanisms available, such as VCG auction or generalized second price auction as desired.

Comparison to SA. The main difference with SA is, similar to what is discussed in the previous subsection, that user satisfaction is a much more important measure to account for. For example, the typical objective in SA (see (1) in Section 1) is social welfare which only accounts for platform and advertisers' utility. In LLMA, however, one might also need to consider user's utility as a function of predicted SR, which would change the allocation function of the mechanism and the payment correspondingly.

3. MARKET PLAYERS AND DESIDERATA

Given the potential framework for LLMA and the presented design choices, we outline several crucial aspects for evaluating each design choice's feasibility and practicality.

In modern search or display ad auctions, numerous market players are involved, including advertisers, users, and platforms. The platform itself is often divided among several players like the demand-side platform, supply-side platform, publisher, and ad exchange. In essence, the ad platform must balance these players' utilities to create a sustainable ecosystem. For example, if the platform shows too many ads in response to a user's query, user retention will likely decrease, discouraging advertisers from using the platform, and eventually harming the ecosystem. Similarly, for LLMA to sustain long-term revenue growth, it must balance the utilities of market players. We will outline the key aspects of the most crucial players: the user, the advertiser, and the platform.

3.1 Player's incentive

User experience. When adding advertisements to LLM output, maintaining high content quality is crucial. Users dislike excessive or irrelevant ads, which can degrade the output and reduce user satisfaction and retention. In modern online ad systems, floor prices are used to filter out irrelevant ads and preserve user experience quality. Similarly, for LLM services, we must ensure that the final output, including ads, remains a high-quality response and closely aligns with what the LLM would originally generate.

Advertiser experience. Advertisers pay the LLMA to include ads in outputs, expecting their products or services to be showcased compellingly. Ads should be engaging and interesting to users, efficiently driving revenue for the advertisers at

[12]This form doesn't allow for adjusting the final output to balance multiple advertisers' preferences, unlike auctions that consider bids and preferences, as in [Duetting et al. 2023].

smaller costs. It is worth noting that advertisements may potentially reduce the overall number of users engaging with the system, which could have adverse effects on the LLMA itself.

Platform revenue. Revenue is LLMA's primary goal, so it must ensure that the additional cost of advertisements is covered by the revenue from advertisers. This is especially critical for LLMA compared to SA, due to the higher computational costs and infrastructure required to run LLMs.

3.2 Desiderata

To balance the utilities of all players and ensure a sustainable LLMA, the following criteria should be considered from the platform's perspective.

Output quality. The LLM's textual output should align with the user's preferences and the query. This involves (a) ensuring the ad is relevant to the user's query and (b) making sure the LLM output aligns with the user's preferences. This includes standard LLM evaluation criteria like accuracy, relevance, coherence, and comprehensiveness.

Additionally, the output should reflect advertisers' preferences, ensuring their ads are integrated as desired and their bids accurately represent their preferences for the query and modified output.

Allocational objective and revenue. As a mediator, LLMA can optimize social welfare to achieve allocational efficiency, i.e., allocate ads to maximize social welfare deterministically or try to maximize its revenue. On the other hand, retrieval-augmented generation (RAG) by [Lewis et al. 2020] that probabilistically retrieves a relevant document from a database of factual documents, is shown to enhance LLM performance by managing ambiguity and factuality, diversifying output, and improving robustness to noise and errors one might want to randomly allocate the ad rather than deterministically to improve the quality of the output. Therefore, LLMA's allocation objective should balance these aspects while ensuring sufficient revenue to avoid operating deficits.

Latency. In LLMA, users expect rapid interactions, similar to search auctions, where prompt responses are typical. Adding advertisements to LLM output introduces some latency, but this should be minimal to avoid disrupting the user experience. The latency requirement for LLMA could be less strict than for search auctions because LLMA generates output word-by-word, whereas search auctions need to retrieve all ads and results immediately upon a query.

Reliability and privacy. LLMA must also address potential risks from advertisers, ensuring system reliability and alignment by considering all possible adversarial behaviors [Hendrycks et al. 2020], *e.g.,* harmful contents or spam links in ads. Additionally, maintaining user privacy is crucial. All user context, information, and data must be kept secure (or encoded) to prevent privacy risks from inadvertent disclosure.

4. CHALLENGES

Recall that our overall framework consists of four modules: modification, bidding, prediction, and auction modules. For each module, we address characteristics,

technical challenges, and research questions relevant to practical implementation and evaluation based on the criteria defined in Section 3.

4.1 Modification module

Challenge: **An advertiser modification model should ensure alignment with user preferences and satisfaction while addressing privacy, reliability, and latency issues.** In the advertiser modification model, LLMA must provide q, X, and C to each advertiser adv_i, which can lead to privacy issues by disclosing user information. Addressing this involves partial or indirect information disclosure, possibly using encryption or differential privacy to protect user data while ensuring high-quality outputs.

Additionally, the model faces reliability issues as advertiser-modified outputs might include illegal or spam content, which may degrade user satisfaction compared to original LLM outputs. LLMA may require an additional module to ensure robustness against such adversarial behavior, but this adds cost of computational resources and latency.

Moreover, increased communication between LLMA and advertisers can raise latency. Therefore, developing efficient protocols is crucial for managing a functional online ad system. The advertiser modification model thus requires novel solutions to address privacy, reliability, and latency concerns.

Challenge: **Effectively reflecting advertiser preferences in the output for LLMA modification model.** The LLMA modification model generally faces fewer privacy, reliability, and latency issues compared to others and focuses on enhancing user experience. However, it may not fully align with advertiser preferences in the output modification process, potentially reducing advertiser satisfaction. Thus, it is important to explore methods for better incorporating advertisers' preferences into the modified output, ensuring that it meets their expectations while still improving the user experience.

Prospect: **Balancing the trade-off between LLMA and advertiser modification models.** To improve advertiser satisfaction in the LLMA modification model, one approach is to let advertisers submit indirect indicators of their preferences. Specifically, after receiving the query q, the original output X, and possibly context c, the advertiser provides a document Y (or a list of it) reflecting its preferences. LLMA can then use Y as a prompt to generate the modified output as per RAG framework, allowing more flexibility in capturing advertiser preferences. However, this method introduces additional communication costs, potentially increasing latency.

Challenge: **Reducing the computational burden of generating every potential candidate output.** Both the advertiser and LLMA modification models require generating all possible output candidates for each potential ad. In the LLMA modification model, this involves multiple runs of the LLM's token sequence generation for each ad in the auction. Given the high cost of LLM operations, this approach may not scale well as the number of ads on the platform increases.

Prospect: **Ex-ante allocation without generating potential candidate out-**

puts. One feasible approach is to implement a prefiltering process to reduce the number of ads competing by filtering out less relevant ads. Alternatively, the auction can be run without generating every possible candidate output by using a modular component to predict the characteristics of the modified output for each candidate ad.

For instance, features can be extracted from the query and ad text, and semantic distances can be computed based on text similarity. Advertisers should be aware of this indirect measure and bid accordingly, assuming the semantic distance will represent the expected output quality. The mechanism then uses submitted bids and semantic distances to determine the ad and generate the final output, requiring only a single generation for the allocated ad. Note that [Hajiaghayi et al. 2024] takes this approach.

4.2 Bidding module

Challenge: **Implementing dynamic bidding model without privacy and latency issue.** The main advantage of the dynamic bidding model is its potential to increase advertiser satisfaction, as advertisers can adjust their bids after seeing the modified output. This flexibility can be appealing to advertisers with the technical capability to dynamically set bids. However, the dynamic bidding model may introduce privacy issues if LLMA discloses private information to advertisers. It may also lead to additional latency since the entire set of modified outputs must be delivered to advertisers. Reliability concerns are minimal, as only the bid amount is communicated. In contrast, the static bidding model avoids privacy, latency, and reliability issues but may reduce advertiser satisfaction because advertisers cannot adjust their bids based on the modified output.

Future research could focus on the practicality of dynamic bidding, including developing protocols and algorithms that address privacy and latency concerns. One may also investigate whom the ad market would comprise, when there is a possibility that the advertisers hire a proxy agent to submit bids on behalf of them, and how the proxy agent (or advertisers themselves) can optimize bids in such scenario.

Prospect: **Balancing the trade-off between static/dynamic bidding models.** In the static bidding model, improving advertiser satisfaction can be achieved by defining a more flexible static function as the contract between the advertiser and LLMA. Specifically, LLMA could propose a contract where bids are determined by an indirect measure of the modified output. For example, LLMA and the advertiser might agree on a contract where the bid is inversely proportional to the similarity distance d between the original output X and the modified output X_i. If X_i significantly deviates from X, the similarity distance will be large, leading to a lower bid from the advertiser due to concerns about user experience.

Advertisers will need to understand how LLMA estimates and defines this distance measure. LLMA could also develop a more refined method for assessing user interest, attention, and relevance for the modified output from the advertiser's perspective, allowing advertisers to choose contracts based on their preferences.

4.3 Prediction module

Challenge: **Efficient and precise implementation of prediction module.** Estimating CTR in LLMA can follow principles similar to those in modern online advertising. We can train a prediction system to estimate $\text{ctr}_i \in [0, 1]$ based on input X_i, q, and c, using user feedback data. For example, factorization machines and online algorithms can be used after feature extraction from user data, context, and queries, as described in [McMahan et al. 2013]. Given the sparse frequency of each X_i in the family of possible outputs, features from q, X_i, and c should be extracted to map to CTR values. User actions like regenerating responses, clicking ads, or exiting the LLM can be used to refine the prediction module.

Challenge: **Relevance/similarity distance measure to estimate user satisfaction.** To estimate SR, one approach is to assume that the original output X is optimal. In this case, the distance between X and the modified output X_i can serve as a measure of SR, since the closer X_i is to X, the higher the expected user satisfaction. For example, one might define the output distance as $d(X, X_i) := ||\Pr(X|q) - \Pr(X_i|q)||$ using a suitable norm. Here, $\Pr(X|q)$ represents the marginal probability of X given q, which can be computed using standard methods from the literature [Vaswani et al. 2017].

More general functions, such as semantic similarity between documents [Mikolov et al. 2013; Cer et al. 2018; Conneau et al. 2019], can be used. One can further implement a calibration layer to ensure it remains well-calibrated with higher accuracy [McMahan et al. 2013]. The key research question is to identify effective measures for predicting user satisfaction when ads are incorporated into the output.

Prospect: **Incorporating distance measures and online learning.** One may consider combining similarity measures and online learning from user feedback for prediction. To learn online SR (or CTR) estimates from user feedback, a useful indicator of whether the user is satisfied with the output is whether the user *regenerates* the output. One may aim to learn a function which outputs the sr, given the query q, modified output X', and context c. This approach does not assume that the original output X is indeed optimal, thereby allowing the possibility that the user may be satisfied with a modified output X_i even though its distance from X is measured is large. This comes at the cost of additional modular component for learning process. One could further consider an online learning model where the similarity distance is also integrated as one of the features for prediction. If the similarity distance has some positive correlations with the true user satisfaction rate, this would increase the accuracy of the prediction. Essentially, an effective way to capture the both advantages of online learning and distance measure should be studied thoroughly.

4.4 Auction module

Challenge: **Incorporating multiple ads in a single output.** Recall that our framework is presented for a setting where a single ad is allocated. One approach to generalize our framework is to repeatedly run the overall procedure and allocate a single ad at once. That is, one can determine an abstract unit that partitions the output, *e.g.,* paragraph, and run our framework for each unit. Another direct approach to extend our framework is, to let the final displayed output X' not

necessarily belong to $\{X_i\}_{i\in[n]}$, but rather interpolates $\{X_i\}_{i\in[n]}$ by prompting LLM to generate output that simultaneously advertises multiple ads. This resembles the approach of aggregating the preference by [Duetting et al. 2023; Soumalias et al. 2024]. By doing so, it might be possible to deliver multiple advertisements in a fair manner, thereby allowing LLMA to bring more revenue by charging multiple advertisers at once.

One subtle issue is that, since each advertiser bids b_i on delivering X_i, they may not want to write the same bid for the balanced output X', thus it may degrade the advertiser's experience. In the static bidding model, as discussed in Section 4.2, this might be handled by committing to a contract based on measures that represents the advertiser's preferences more in a refined manner. In the dynamic bidding model, one approach would be to append an additional step of asking for bids for the final output again to the advertisers.

4.5 Discussion on Recent Approaches

Several recent theoretical approaches have proposed game-theoretic models for ad auctions in LLMs. We discuss how these can be viewed as implementations of our framework, facilitating future research through modular comparisons of components in each approach and highlighting their pros and cons more explicitly.

[Duetting et al. 2023] propose a model where bidders submit bids and distributions over tokens. This aligns with a tokenized version of our model with (i) LLMA modification by aggregating token distributions, (ii) dynamic bidding with adjustments for each query, (iii) no prediction module as it focuses on advertiser perspectives, and (iv) running a token auction. This approach might suffer from issues of latency and reliability, but could possibly maintain high advertiser experience.

[Hajiaghayi et al. 2024] integrate the RAG framework for segment-based ad auctions, where ads are probabilistically retrieved for segments like paragraphs using bids and a notion of relevance. This corresponds to (i) LLMA modification without pre-generating outputs, (ii) static bidding based on clicks, (iii) indirect CTR prediction by measuring ad relevance, and (iv) running a segment auction. Their approach has less privacy, latency and reliability, but advertiser's experience could be largely dependent on how modular components to compute relevance are designed.

[Dubey et al. 2024] introduce the concept of a prominence auction, wherein the allocation function determines both the prominence of each selected advertisement in the output and the specific ads to be allocated. The prominence assigned to each ad influences its representation in the LLM-generated summary, thereby affecting user attention and engagement. This is shown to be generalizing the standard position auction by [Varian 2007]. This aligns with (i) LLMA modification, (ii) static bidding, (iii) no prediction required, as prominence serves as a CTR proxy, and (iv) running a prominence auction.Similar to [Hajiaghayi et al. 2024], the advertiser's experience highly depends on how well the LLM in the modification module constructs the output with desired prominence.

[Soumalias et al. 2024] propose an auction that truthfully aggregates advertiser preferences using reinforcement learning from human feedback (RLHF), widely used in LLMs to align the outputs of LLMs with diverse human preferences. This can

be viewed as (i) LLMA modification, (ii) static bidding, (iii) no prediction needed, focusing only on advertiser perspectives, and (iv) running an RLHF-based auction. This creates greater issues with latency, since every potential candidate output needs to be created and the advertiser's reward function should be communicated accordingly. However, it could better reflect advertiser's preferences.

5. CONCLUDING REMARKS

There are several potential areas beyond those outlined here. As observed in most modern online advertising platforms, the use of autobidders [Aggarwal et al. 2024], which delegate the bidding process to the platform, appears to be a plausible approach. Another promising direction is leveraging LLMs themselves to enhance modern search and display advertising systems by efficiently tailoring ad content to individual users. For instance, LLMs could be integrated into the standard dynamic creative optimization framework, often referred to as responsive advertising [Google 2024]. For more discussions of these perspectives, we refer to the full paper [Feizi et al. 2023].

REFERENCES

Abd-Alrazaq, A. A., Rababeh, A., Alajlani, M., Bewick, B. M., and Househ, M. 2020. Effectiveness and safety of using chatbots to improve mental health: systematic review and meta-analysis. *Journal of medical Internet research 22,* 7, e16021.

AdWeek. 2023. Microsoft advertisements on AI driven chat based search. `https://www.adweek.com/media/microsoft-details-how-advertising-works-on-bings-ai-driven-chat-based-search`.

Aggarwal, G., Badanidiyuru, A., Balseiro, S. R., Bhawalkar, K., Deng, Y., Feng, Z., Goel, G., Liaw, C., Lu, H., Mahdian, M., et al. 2024. Auto-bidding and auctions in online advertising: A survey. *ACM SIGecom Exchanges 22,* 1, 159–183.

Anil, R., Dai, A. M., Firat, O., Johnson, M., Lepikhin, D., Passos, A., Shakeri, S., Taropa, E., Bailey, P., Chen, Z., et al. 2023. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems 33*, 1877–1901.

Cer, D., Yang, Y., Kong, S.-y., Hua, N., Limtiaco, N., John, R. S., Constant, N., Guajardo-Cespedes, M., Yuan, S., Tar, C., et al. 2018. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*.

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., and Stoyanov, V. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Crunch, T. 2023. That was fast! Microsoft slips ads into AI-powered Bing Chat. `https://techcrunch.com/2023/03/29/that-was-fast-microsoft-slips-ads-into-ai-powered-bing-chat/`.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Dubey, K. A., Feng, Z., Kidambi, R., Mehta, A., and Wang, D. 2024. Auctions with llm summaries. *arXiv preprint arXiv:2404.08126*.

Duetting, P., Mirrokni, V., Leme, R. P., Xu, H., and Zuo, S. 2023. Mechanism design for large language models. *arXiv preprint arXiv:2310.10826*.

Edelman, B., Ostrovsky, M., and Schwarz, M. 2007. Internet advertising and the generalized second-price auction: Selling billions of dollars worth of keywords. *American economic review 97,* 1, 242–259.

Feizi, S., Hajiaghayi, M., Rezaei, K., and Shin, S. 2023. Online advertisements with llms: Opportunities and challenges. *arXiv preprint arXiv:2311.07601*.

Fried, D., Aghajanyan, A., Lin, J., Wang, S., Wallace, E., Shi, F., Zhong, R., Yih, W.-t., Zettlemoyer, L., and Lewis, M. 2022. Incoder: A generative model for code infilling and synthesis. *arXiv preprint arXiv:2204.05999*.

Goel, S., Lahaie, S., and Vassilvitskii, S. 2009. Contract auctions for sponsored search. In *International Workshop on Internet and Network Economics*. Springer, 196–207.

Google. 2024. About responsive search ads . `https://https://support.google.com/google-ads/answer/7684791?hl=en`.

Guerreiro, N. M., Alves, D., Waldendorf, J., Haddow, B., Birch, A., Colombo, P., and Martins, A. F. T. 2023. Hallucinations in large multilingual translation models.

Hajiaghayi, M., Lahaie, S., Rezaei, K., and Shin, S. 2024. Ad auctions for llms via retrieval augmented generation. *arXiv preprint arXiv:2406.09459*.

Hendrycks, D., Burns, C., Basart, S., Critch, A., Li, J., Song, D., and Steinhardt, J. 2020. Aligning ai with shared human values. *arXiv preprint arXiv:2008.02275*.

Insight, M. 2022. Global Search advertising. `https://www.millioninsights.com/snapshots/search-advertising-market-report`.

Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y. J., Madotto, A., and Fung, P. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys 55,* 12 (Mar.), 1–38.

Kirk, H. R., Jun, Y., Volpin, F., Iqbal, H., Benussi, E., Dreyer, F., Shtedritski, A., and Asano, Y. 2021. Bias out-of-the-box: An empirical analysis of intersectional occupational biases in popular generative language models. In *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds. Vol. 34. Curran Associates, Inc., 2611–2624.

Lahaie, S., Pennock, D. M., Saberi, A., and Vohra, R. V. 2007. Sponsored search auctions. *Algorithmic game theory 1*, 699–716.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems 33*, 9459–9474.

Li, J., Cheng, X., Zhao, W. X., Nie, J.-Y., and Wen, J.-R. 2023. Halueval: A large-scale hallucination evaluation benchmark for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. 6449–6464.

Liang, P. P., Wu, C., Morency, L.-P., and Salakhutdinov, R. 2021. Towards understanding and mitigating social biases in language models. In *Proceedings of the 38th International Conference on Machine Learning*, M. Meila and T. Zhang, Eds. Proceedings of Machine Learning Research, vol. 139. PMLR, 6565–6576.

Liu, H., Li, C., Wu, Q., and Lee, Y. J. 2023. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*.

Liu, W., Wang, X., Wu, M., Li, T., Lv, C., Ling, Z., Zhu, J., Zhang, C., Zheng, X., and Huang, X. 2023. Aligning large language models with human preferences through representation engineering.

Liu, Y., Yao, Y., Ton, J.-F., Zhang, X., Guo, R., Cheng, H., Klochkov, Y., Taufiq, M. F., and Li, H. 2024. Trustworthy llms: a survey and guideline for evaluating large language models' alignment.

McMahan, H. B., Holt, G., Sculley, D., Young, M., Ebner, D., Grady, J., Nie, L., Phillips, T., Davydov, E., Golovin, D., et al. 2013. Ad click prediction: a view from the trenches. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. 1222–1230.

Microsoft. 2023. Monetize AI powered chat experiences. `https://about.ads.microsoft.com/en-us/blog/post/may-2023/a-new-solution-to-monetize-ai-powered-chat-experiences`.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Navigli, R., Conia, S., and Ross, B. 2023. Biases in large language models: origins, inventory, and discussion. *ACM Journal of Data and Information Quality 15,* 2, 1–21.

Netflix. 2022. Netflix Started Ad-supported plan. `https://help.netflix.com/en/node/126831/`.

Nicolescu, L. and Tudorache, M. T. 2022. Human-computer interaction in customer service: the experience with ai chatbots—a systematic literature review. *Electronics 11,* 10, 1579.

Nijkamp, E., Pang, B., Hayashi, H., Tu, L., Wang, H., Zhou, Y., Savarese, S., and Xiong, C. 2022. Codegen: An open large language model for code with multi-turn program synthesis. *arXiv preprint arXiv:2203.13474*.

Outbrain. 2023. Native is better than display. `https://www.outbrain.com/blog/native-ads-vs-display-ads/`.

PCMag. 2023. LLM is replacing search engine. `https://www.pcmag.com/news/when-will-chatgpt-replace-search-engines-maybe-sooner-than-you-think`.

Roughgarden, T. 2010. Algorithmic game theory. *Communications of the ACM 53,* 7, 78–86.

Shen, T., Jin, R., Huang, Y., Liu, C., Dong, W., Guo, Z., Wu, X., Liu, Y., and Xiong, D. 2023. Large language model alignment: A survey. *arXiv preprint arXiv:2309.15025*.

Soumalias, E., Curry, M. J., and Seuken, S. 2024. Truthful aggregation of llms with an application to online advertising. *arXiv preprint arXiv:2405.05905*.

Thoppilan, R., De Freitas, D., Hall, J., Shazeer, N., Kulshreshtha, A., Cheng, H.-T., Jin, A., Bos, T., Baker, L., Du, Y., et al. 2022. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*.

Varian, H. R. 2007. Position auctions. *international Journal of industrial Organization 25,* 6, 1163–1178.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems 30*.

Wang, B. and Komatsuzaki, A. 2021. Gpt-j-6b: A 6 billion parameter autoregressive language model.

Weidinger, L., Mellor, J., Rauh, M., Griffin, C., Uesato, J., Huang, P.-S., Cheng, M., Glaese, M., Balle, B., Kasirzadeh, A., Kenton, Z., Brown, S., Hawkins, W., Stepleton, T., Biles, C., Birhane, A., Haas, J., Rimell, L., Hendricks, L. A., Isaac, W., Legassick, S., Irving, G., and Gabriel, I. 2021. Ethical and social risks of harm from language models.

Zhang, Y., Li, Y., Cui, L., Cai, D., Liu, L., Fu, T., Huang, X., Zhao, E., Zhang, Y., Chen, Y., et al. 2023. Siren's song in the ai ocean: a survey on hallucination in large language models. *arXiv preprint arXiv:2309.01219*.