informs
https://pubsonline.informs.org/journal/mksc

MARKETING SCIENCE
Vol. 44, No. 5, September–October 2025, pp. 995–1016
ISSN 0732-2399 (print), ISSN 1526-548X (online)

LOLA: LLM-Assisted Online Learning Algorithm for Content Experiments

Zikun Ye,[a,*] Hema Yoganarasimhan,[a,*] Yufeng Zheng[b]

[a] University of Washington, Seattle, Washington 98195; [b] University of Toronto, Toronto, Ontario M5S 1A1, Canada
*Corresponding authors

Contact: zikunye@uw.edu, https://orcid.org/0000-0001-9914-7966 (ZY); hemay@uw.edu, https://orcid.org/0000-0003-0703-5196 (HY); yufeng.zheng@mail.utoronto.ca, https://orcid.org//0009-0004-4125-9446 (YZ)

Received: July 25, 2024
Revised: November 26, 2024; January 29, 2025
Accepted: February 11, 2025
Published Online in Articles in Advance: March 24, 2025

https://doi.org/10.1287/mksc.2024.0990

Abstract. Modern media firms require automated and efficient methods to identify content that is most engaging and appealing to users. Leveraging a large-scale data set from Upworthy (a news publisher), which includes 17,681 headline A/B tests, we first investigate the ability of three pure–large language model (LLM) approaches to identify the catchiest headline: prompt-based methods, embedding-based methods, and fine-tuned open-source LLMs. Prompt-based approaches perform poorly, while both OpenAI embedding–based models and the fine-tuned Llama-3-8B achieve marginally higher accuracy than random predictions. In sum, none of the pure LLM–based methods can predict the best-performing headline with high accuracy. We then introduce the LLM-assisted online learning algorithm (LOLA), a novel framework that integrates LLMs with adaptive experimentation to optimize content delivery. LOLA combines the best pure-LLM approach with the upper confidence bound algorithm to allocate traffic and maximize clicks adaptively. Our numerical experiments on Upworthy data show that LOLA outperforms the standard A/B test method (the current status quo at Upworthy), pure bandit algorithms, and pure-LLM approaches, particularly in scenarios with limited experimental traffic. Our approach is scalable and applicable to content experiments across various settings where firms seek to optimize user engagement, including digital advertising and social media recommendations.

History: Olivier Toubia served as the senior editor.
Supplemental Material: The online appendix and data files are available at https://doi.org/10.1287/mksc.2024.0990.

Keywords: LLMs • content experiments • bandits • news • digital marketing

1. Introduction

Digital content consumption has seen unprecedented growth, leading to a proliferation of content across various platforms. In today's real-time digital environment, media firms and news publishers need automated and efficient methods to determine which content generates high user engagement and platform growth. This includes identifying the most appealing articles, the most attractive headlines, and the catchiest cover images (Symonds 2017, Coenen 2019). Traditionally, media firms and publishers have relied on experimentation-based approaches to address this problem. Broadly speaking, there are two types of experimentation styles: (1) standard A/B tests, and (2) online learning algorithms or bandits.[1]

The A/B test is the most straightforward experimentation method, where a firm allocates a fixed portion of traffic to different treatment arms, assesses the results, and then goes with the best-performing arm. This is also known as the explore-and-commit strategy (E&C), where the firm explores for a fixed period using A/B tests and then commits to one treatment based on the results of those tests. This approach is widely used in the publishing industry. For instance, the *New York Times* recently built a centralized internal A/B test platform, A/B Reporting and Allocation architecture (ABRA), allowing different teams to experiment with their content; see Yang (2024) for more details. Another example is Upworthy, which has implemented extensive A/B tests to choose headlines for news articles (Matias et al. 2021). The main advantage of this approach is that it is trustworthy: with a sufficiently large amount of traffic allocated to the A/B test, the resulting inference is unbiased and accurate. However, a major drawback of this approach is the wastage of traffic; by assigning equal traffic to all the treatment arms during the exploration phase (including the poorly performing ones), the firm incurs high regret or low profits. This is especially problematic in the news industry because news articles tend to have short lifetimes and become stale quickly (typically within a day or two). Therefore, if a firm wastes a lot of traffic to learn the best article/headline, the article itself might become irrelevant by the end of the A/B test.

The second approach, online learning algorithms, or bandits, are able to address some of these problems. These algorithms dynamically adjust traffic allocation to articles/headlines based on their features, past performance, and associated uncertainty. This method provides a more efficient way to optimize content delivery by continually learning and adapting. For instance, both Yahoo News and the *New York Times* have successfully implemented contextual bandit algorithms to identify content that maximizes user engagement (Li et al. 2010, Coenen 2019). Online learning algorithms typically outperform A/B tests in terms of overall regret or reward. These algorithms reduce the extent of exploration over time by moving traffic away from poorly performing treatments dynamically. As a result, they tend to switch to the best-performing arms quickly and incur lower regret compared with standard E&C strategies. This has been established both theoretically (Lattimore and Szepesvári 2020) as well empirically (Li et al. 2010).[2]

Nevertheless, online algorithms also have notable drawbacks. In standard-content experiments, these algorithms suffer from a cold-start problem. That is, firms typically start these adaptive experiments assuming that all arms are equally likely to be the best-performing ones. This can lead to a situation where more-than-necessary traffic is allocated to suboptimal arms, leading to higher regret. Additionally, because of their probabilistic nature, these algorithms can inadvertently converge on a suboptimal arm, especially when the experimental traffic is small. These issues arise because online algorithms are designed for large traffic regimes, where the initialization of arms has little impact on the asymptotic regret. However, as discussed earlier, there is limited traffic available for experimentation in the media industry because content becomes stale quickly. Therefore, the negative effects of the "equally effective" initialization do not diminish sufficiently during the time horizon of interest, leading to higher regret.

Given these issues with content experimentation, the main question that this paper asks and answers is whether we can leverage the recently developed large language models (LLMs) to enhance current content experimentation practices and simplify the process of identifying appealing content. LLMs, trained on extensive human-generated data, have demonstrated the ability to mimic human preferences and behavior in a variety of consumer research tasks (Brand et al. 2023, Li et al. 2024). Specifically, in the context of news articles, Yoganarasimhan and Yakovetskaya (2024) show that LLMs can correctly predict the polarization of news content. Hence, a relevant question is whether firms can directly use LLMs to predict the appeal of different content, potentially replacing traditional experimentation. This approach could significantly reduce the costs associated with experimentation while simplifying the content delivery processes.

To address this question, we employ a large data set of content experiments by the publishing firm Upworthy conducted in the 2013–2015 time frame. The data set consists of 17,618 headline tests spanning 77,245 headlines, 277 million impressions, and 3.7 million clicks. The goal of each A/B test was to identify the best headline among a set of candidate headlines (for a given article). This data set is an excellent test bed for our study for a few reasons: (1) it is based on an extremely large number of A/B tests over a high volume of impressions; (2) each A/B test received a relatively large number of impressions, allowing for accurate estimates of each headline's performance; and (3) it is sourced from a real media firm and represents the actions of a very large number of readers over a sufficiently long period to offer meaningful insights.

In the first part of the paper, we use this data set to examine the extent to which pure LLM–based approaches can accurately predict headline attractiveness. For this exercise, we test the different approaches' accuracies, defined as the success rate of identifying the headline with the highest click-through rate (CTR) in a given A/B test.

We consider the three most widely used LLM-based approaches for this analysis: (1) prompt-based approaches, (2) LLM text-embedding approaches, and (3) fine-tuning approaches. For prompt-based methods, we consider two approaches: (1) zero-shot prompting, and (2) in-context learning. Zero-shot prompting is a technique in which an LLM generates responses without being explicitly trained on specific examples. This is the most common way in which most users interact with the LLM. In-context learning involves providing the LLM with a few demonstrations of similar tasks before generating responses for a specific task (Xie et al. 2021, Min et al. 2022). We find that GPT-3.5 performs as poorly as a random guess for both prompt-based approaches. GPT-4 performs marginally better—it achieves an accuracy of 37.85% for zero-shot prompting and 38%–40% for in-context learning. Overall, this suggests that prompt-based approaches cannot aid/replace content experiments in any meaningful way.

Next, we turn to the second approach: text embeddings. Specifically, we transform headlines into embedding vectors using the most recent embedding model from OpenAI. We then use the embedding as the input, along with the CTR as the output, to train CTR prediction models, including linear regression and multilayer perceptron (a type of neural network). After getting predicted CTRs, we choose the headline with the highest predicted CTR as the winner. We find that OpenAI's embeddings combined with simple linear regression can achieve around 46.28% accuracy, whereas more complex neural networks do not improve performance over the simple linear model. Finally, we fine-tune a state-of-the-art open-source LLM, Llama-3 with eight billion parameters, using low-rank adaptation (LoRA) (Hu et al. 2022). This approach performs the best, with 46.86% accuracy, which is a small improvement over the embedding-based approach. However, even the best fine-tuned LLMs are unable to perfectly predict which types of content are most appealing to users. As such, none of the LLM-based approaches can match the accuracy of experimentation-based approaches.

Therefore, in the second part of the paper, we propose and evaluate a novel framework for content experiments that combines the strengths of LLM-based approaches with the advantages of online learning algorithms, termed LLM-assisted online learning algorithm (LOLA). Our main insight is to use predictions from an LLM model as priors in online algorithms to optimize experimentation. Whereas the general idea that priors can impact regret and rewards has existed for some time (Bubeck and Liu 2013, Russo and Van Roy 2014), these papers typically assume that the quality and distribution of priors are known. However, a key challenge in our setting is that the LLM predictions cannot be naively treated as priors because we do not know their prediction error or theoretical properties. Therefore, we build on the recently proposed 2-upper confidence bounds (2-UCBs) algorithm by Gur and Momeni (2022) to overcome this challenge. The original 2-UCBs algorithm was designed for a setting where the experimenter has access to auxiliary data and knows both the size of this auxiliary sample (n^{aux}) and knowledge of the prediction error in this sample. Our innovation is to extend this algorithm to accommodate LLM predictions by treating these predictions *as if* they came from a "pseudosample" and then to fine-tune the size of this pseudosample for a given application based on prior data using hyperparameter tuning. Thus, we are able to combine LLM predictions with online experimentation without making strong assumptions on the theoretical properties of the LLM-based predictors.

LOLA involves two steps. First, we train an LLM-based CTR prediction model using LoRA fine-tuning. The second step integrates these predictions into online learning algorithms. LOLA is a general framework, and depending on the specific goals of the experimentation, it can be adapted for different prediction models in the first step and different online algorithms in the second step. For example, if the goal is to minimize regret (i.e., maximize total reward/clicks), we propose a modified upper confidence bound algorithm called LLM-assisted 2-upper confidence bounds (LLM-2UCBs), which is detailed in Section 4.2. The idea of LLM-2UCBs is intuitive: we can view LLM-based CTR predictions as auxiliary samples before the start of the online algorithm (e.g., a 1% CTR can be viewed as 100 impressions with one click or 1,000 impressions with 10 clicks). After fine-tuning this auxiliary impression–size hyperparameter, we incorporate these auxiliary samples into the standard UCB algorithm (Auer et al. 2002) to obtain the reward estimator and shrink the upper confidence bound.

LOLA combines the advantages of both experimentation and LLMs, making it accurate and efficient. By integrating an LLM-based model to predict CTRs before the online deployment phase, we avoid wasting impressions on poorly performing headlines, especially in the initial stage when we face a cold-start problem. On the other hand, through experimentation, we can correct the errors in LLM predictions. This hybrid approach ensures that whereas LLMs provide reasonable initial predictions, ongoing experimentation refines and improves overall performance, making LOLA an effective solution for optimizing content delivery.

We present a comprehensive evaluation of LOLA and compare its performance with three natural benchmarks using the Upworthy data set: (1) explore and commit, (2) a pure online learning algorithm, and (3) a pure LLM-based model. The first benchmark, E&C, was the status quo at Upworthy during the time of data collection, wherein the editors first ran an A/B test on a set of headlines for a fixed set of impressions and then displayed the winner for the rest of the traffic. For the pure online learning algorithm, we use the standard UCB algorithm, which is initialized with uniform CTRs. For the pure LLM–based approach, we use the LoRA fine-tuned Llama-3 to predict the CTRs of headlines and select the one with the highest CTR for all impressions, bypassing the experimentation phase. In LOLA, we use the same CTR prediction model as that used in the pure-LLM approach and then employ LLM-2UCBs to maximize accumulated clicks. We find that LOLA outperforms all the baseline approaches, though the next-best algorithm varies based on the length of the time horizon (keeping the number of headlines fixed).

Specifically, when time horizons are small, both LOLA and the pure LLM–based approach beat the pure experimentation-based approaches (i.e., E&C and the UCB algorithm) because of relatively accurate LLM-based CTR predictions and lack of sufficient time for the experimentation to provide meaningful updates. Specifically, both LOLA and the pure LLM outperform E&C by 8%–9% and UCB by 6%–7% in this scenario. However, as the number of impressions/time horizon grows, the performance of the pure LLM–based approach falls behind other experimentation-based methods because pure LLM–based approaches are static and do not take advantage of experimentation. In this case, LOLA beats pure-LLM methods and E&C by 4%–5% and UCB by

2%–3%. In sum, our experiments demonstrate that LOLA is able to leverage the power of LLMs early in the horizon and then build on it to take advantage of experimentation over time, thereby bringing together the strengths of both methods.

Finally, whereas we focus on regret minimization in a stochastic bandit setting given the Upworthy context, LOLA is a general-purpose framework and can easily accommodate different goals and/or settings. We provide variants of the standard LOLA (and results on the empirical performance of these variants when applicable) for a few natural extensions, for example, for Bayesian bandit settings where the second-stage algorithm can be Thompson sampling instead of UCB, for the best arm identification (BAI) problem where the goal is to identify the best content/headline rather than regret minimization, and for settings where the firm has access to user/contextual features that can be used to personalize content delivery.

In summary, our paper makes three key contributions to the literature. First, from an empirical perspective, the paper provides evidence that LLMs (or, more precisely, the current vintage of LLMs) cannot replace experimental approaches in the media and digital marketing industry. Second, from a methodological perspective, we develop a novel framework, LOLA, to leverage LLMs for improving content experimentation in digital settings. To the best of our knowledge, this is the first paper that proposes combining LLMs with adaptive experimentation techniques. We demonstrate the value of our approach using a large-scale data set from Upworthy when compared with baseline methods. Our work thus provides a foundation for future research into more advanced algorithms and offers insights for real-world validation in industry settings. Third, from a managerial perspective, LOLA can be used in a broad range of settings where firms need to decide which content to show. Whereas we focus on the news/publishing industry, given our empirical context, the LOLA framework is general and can easily extend to other settings, including digital advertising, email marketing, and website design. Further, the method is cost-effective, can be built on open-source LLMs (alleviating data privacy concerns), and is easily deployable across different domains once fine-tuned on relevant data sets.

2. Upworthy Data and Experiments

We now discuss the data sourced from Upworthy, a U.S. media publisher known for its extensive use of A/B tests in digital publishing. Upworthy conducted randomized experiments with each article published, exploring different combinations of headlines and images to determine which elements most effectively led to higher clicks. The archival record from January 24, 2013, to April 30, 2015, detailed in Matias et al. (2021), demonstrates how the packaging of headlines and images played a crucial role in Upworthy's growth strategy. Upworthy's editorial team created several versions of headlines and/or images for each article (internally called "package" by Upworthy). A package is defined as one treatment or arm for an article and consists of a headline, image, or a combination of both. Editors would first choose several of what they believed were the most promising packages to be tested. Then, they A/B tested these packages or treatments to identify which one resonated the best with their audience. During the A/B test, users only saw the headline (and an accompanying image in some cases) but not the article itself. Upon clicking on the headline, they were taken to the article. As such, the content of the article itself did not have any effect on users' clicking behavior. Upworthy's A/B test system recorded how many impressions and clicks each package received. Table 1 shows two examples of A/B tests with the relevant columns. The full details of all the A/B tests and their results are available from the Upworthy Research Archive. This data set has also been utilized in marketing research about headline engagement evaluations; for example, Banerjee and Urminsky (2024) investigate how the informational, cognitive, linguistic, and affective factors can affect engagement.

We now describe the details of this data set and discuss our preprocessing procedure. In the original data set, there are 150,817 tested packages from 32,487 deployed A/B tests. The data set records a total of

Table 1. Samples of Upworthy Experiments' Results for Headlines

Test ID	Headline	Impressions	Clicks
1	New York's Last Chance To Preserve Its Water Supply	2,675	15
1	How YOU Can Help New York Stay Un-Fracked in Under 5 Minutes	2,639	19
1	Why Yoko Ono Is the Only Thing Standing Between New York And Catastrophic Gas Fracking	2,734	34
2	If You Know Anyone Who Is Afraid of Gay People, Here's a Cartoon That Will Ease Them Back to Reality	4,155	120
2	Hey Dude. If You Have An Older Brother, There's a Bigger Chance You're Gay	4,080	41

Note. We display the key columns for our analysis, including the ID for the A/B tests, the text of headlines, and the number of impressions and clicks in the test.

Table 2. Basic Summary Statistics of Upworthy Data During the Time Window Between January 24, 2013, and April 30, 2015

Statistics summary	Value
Panel A: Original data	
Number of tests	32,487
Number of packages	150,817
Number of impression in tests	538,272,878
Number of clicks in tests	8,182,674
Panel B: Headline test data	
Number of tests	17,681
Number of packages	77,245
Number of impression in tests	277,338,713
Number of clicks in tests	3,741,517
Number of impressions (both test and nontest impressions)	2,351,171,402

Table 3. Distribution of Number of Headlines Tested Across A/B Tests

Number of headlines in one test	Number of tests	Percent of samples (%)
2	1,619	9.16
3	939	5.31
4	8,836	49.97
5	2,964	16.76
6	2,685	15.19
7 or more	638	3.61

538,272,878 impressions and 8,182,674 clicks. All of these statistics are also listed in panel A of Table 2. These summary statistics indicate that during this period, each A/B test had an average of 4.64 packages, with each package receiving an average of 3,569 impressions and 54.26 clicks, resulting in an average click-through rate (CTR) of 1.52%. Within an A/B test, each package/treatment arm had the same probability of receiving an impression, so the number of impressions received by all the packages within a test is approximately the same. However, the actual number of impressions varies across tests, with the first quartile at 2,745, the median at 3,117, and the third quartile at 4,089. This is because of the relatively straightforward implementation of A/B tests; that is, Upworthy did not conduct any power analysis in advance to determine the traffic for a given test, as confirmed by Matias et al. (2021).

Among these tests, some were conducted to test different images. However, the data set does not allow us to trace back the actual images used in the tests. Therefore, we focus on the headline tests and filter out all the image tests. After this filtering, there are 17,681 tests, totaling 77,245 packages. On average, each article was tested with 4.37 headlines. Note that after filtering, each package or treatment simply refers to a unique headline, and therefore, we use the terms "package" and "headline" interchangeably in the rest of the paper. The distribution of the number of packages/headlines for each article is shown in Table 3. After filtering, the data set includes 277,338,713 impressions and 3,741,517 clicks; see panel B of Table 2. We also report the total number of impressions for the entire Upworthy website as recorded in their data from Google Analytics. We see that the impressions attributable to the tests constitute only 22.9% of the total traffic.

To illustrate that learning which headline is best is a nontrivial task, we conduct a survey where we give human users pairs of headlines from the set of headline pairs that are significantly different from each other and ask them to identify the catchier headline. Even within this set of headline pairs, we find that the respondents' accuracy in the survey was not significantly better than random guessing. This further highlights the fact that the task of identifying engaging content is inherently challenging, even for humans. See Web Appendix Section A for more details of the survey and its results.[3]

3. Pure LLM–Based Methods

We divided the 17,681 headline tests into three folds: 70% for training, 10% for hyperparameter fine-tuning of the algorithms (as discussed in Section 4), and 20% for testing. This sample split is used consistently throughout the remainder of the paper unless stated otherwise. Specifically, in this section, we have 12,376 headline tests as a training set and 3,263 headline tests as a test set.[4]

In the rest of this section, we investigate the performance of three widely used LLM-based approaches: (1) GPT prompt-based approaches, including zero-shot prompting and in-context learning; (2) embedding-based models; and (3) fine-tuning of LLMs.

3.1. Prompt-Based Approaches

Prompt-based approaches use GPT prompts to select which headline is the catchiest, and there is no modeling or explicit fine-tuning involved. Prompt-based approaches have been the standard use case for AI models in the marketing literature so far and have been used to examine if and when LLMs can simulate consumer behavior in market research studies (Brand et al. 2023, Gui and Toubia 2023, Li et al. 2024). We consider two prompt-based approaches: (1) zero-shot prompting, and (2) in-context learning.

3.1.1. Zero-Shot Prompting. Zero-shot prompting is a technique in which an LLM generates responses or performs tasks without being explicitly provided by any specific examples. We evaluate the performance using zero-shot prompting based on OpenAI's GPT models. We do this through OpenAI's API in Python, which offers various parameters to customize and control the model's behavior and responses. The key parameters include specifying which GPT model to use (such as GPT-3 or GPT-4) and the system parameters, which guide the type of responses generated. The user's input or query is contained in the user parameter, whereas the assistant parameter holds the model's previous responses to maintain context in an ongoing conversation. Additionally, we set the temperature parameter to zero, which indicates there is no randomness in the GPT's response for reproducibility.[5]

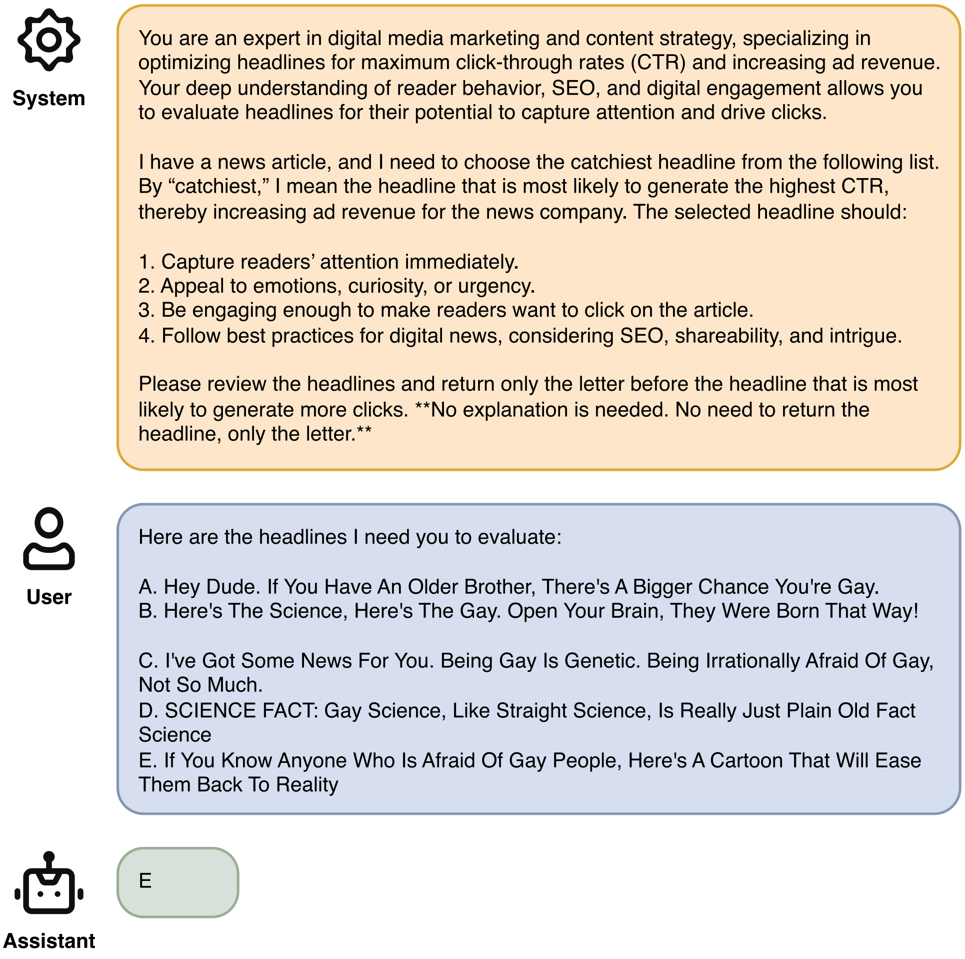
We design the zero-shot prompting to ask GPT which headline is the catchiest, as shown in Figure 1. We tell ChatGPT that it is an expert in marketing and specializing in optimizing headline selection. In this figure, we use letters (`A,B,C, …`) to mark different headlines. Brucks and Toubia (2023) show that different marker types and the order of options provided to LLM might affect the model's response in the sense that LLM might prefer to select the first option or select a certain marker. To mitigate this issue, we also consider other markers, numbers (`1,2,3, …`) and symbols (`!,@,#, …`), and we randomize the order of the options. Further, we adopt the common practice of LLM prompting for a specialized task by detailing the task for the LLM and clearly defining the meaning of a catchy headline. Notably, the first paragraph of the prompt assigns the role of a digital media expert to the LLM. This technique, commonly referred to as role prompting, involves explicitly assigning a specific role to the LLM to guide its responses. Role prompting has been shown to be an effective strategy in the broader survey of prompt engineering techniques in Schulhoff et al. (2024), and we find that it works well in our setting, too.

We test this prompt using three OpenAI large language models,[6] each differing in parameter size: (1) GPT-3.5 (`GPT-3.5-turbo-0125`), which uses around 175 billion parameters; (2) GPT-4 (`GPT-4-turbo-2024-04-09`), which uses approximately 1.7 trillion parameters; and (3) GPT-4o (`GPT-4o-2024-05-13`), which uses an even larger number of parameters. The goal here is to investigate whether improvements in model architecture and parameter size can improve the accuracy of our predictions via standard prompts.

3.1.2. In-Context Learning.

In-context learning is a technique where a model is provided with demonstrations within the input prompt to help it perform a task.

Figure 1. (Color online) Zero-Shot Prompting for Headline Selection



You are an expert in digital media marketing and content strategy, specializing in optimizing headlines for maximum click-through rates (CTR) and increasing ad revenue. Your deep understanding of reader behavior, SEO, and digital engagement allows you to evaluate headlines for their potential to capture attention and drive clicks.

I have a news article, and I need to choose the catchiest headline from the following list. By "catchiest," I mean the headline that is most likely to generate the highest CTR, thereby increasing ad revenue for the news company. The selected headline should:

1. Capture readers' attention immediately.
2. Appeal to emotions, curiosity, or urgency.
3. Be engaging enough to make readers want to click on the article.
4. Follow best practices for digital news, considering SEO, shareability, and intrigue.

Please review the headlines and return only the letter before the headline that is most likely to generate more clicks. **No explanation is needed. No need to return the headline, only the letter.**

Here are the headlines I need you to evaluate:

A. Hey Dude. If You Have An Older Brother, There's A Bigger Chance You're Gay.
B. Here's The Science, Here's The Gay. Open Your Brain, They Were Born That Way!

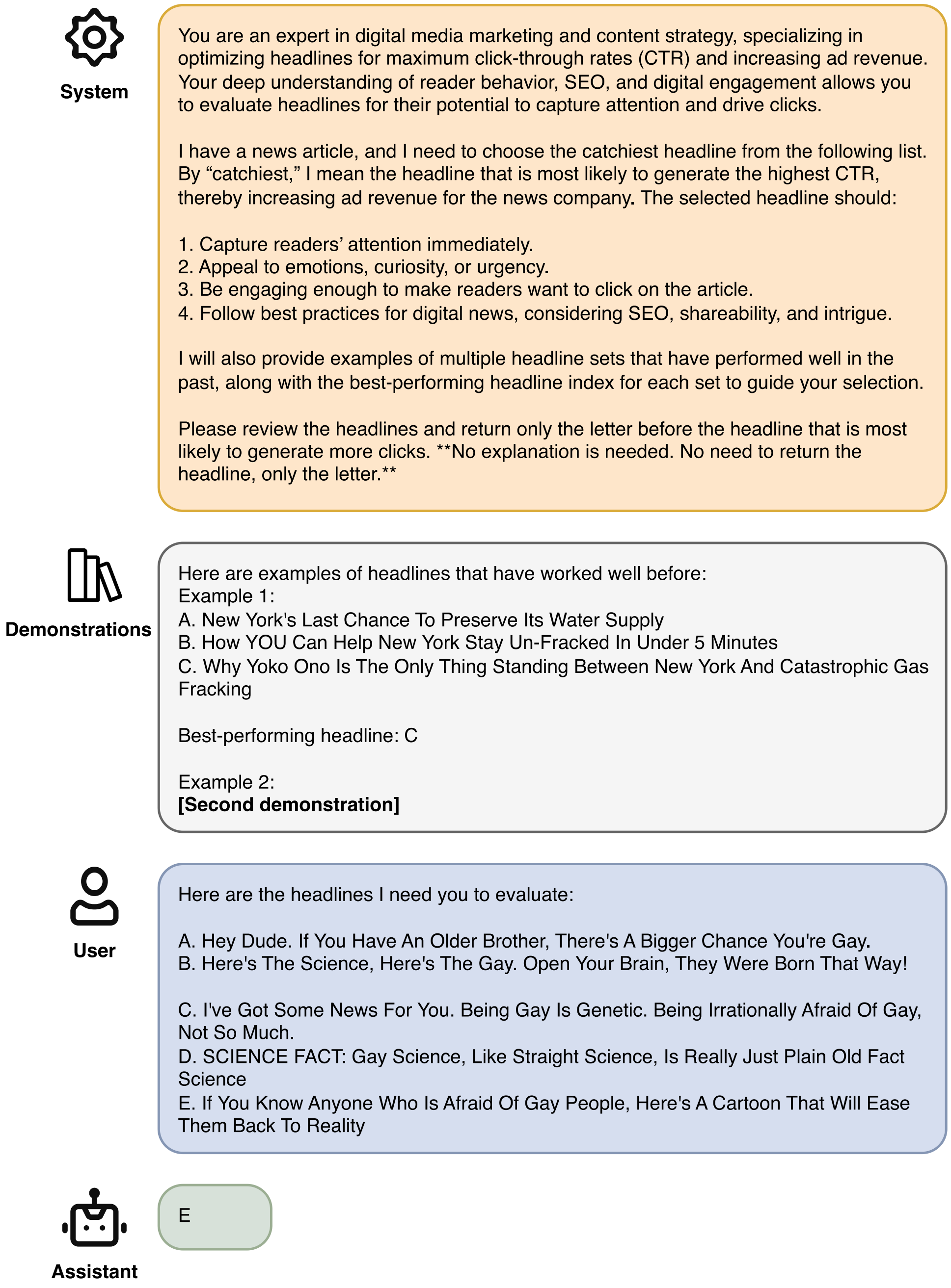
C. I've Got Some News For You. Being Gay Is Genetic. Being Irrationally Afraid Of Gay, Not So Much.
D. SCIENCE FACT: Gay Science, Like Straight Science, Is Really Just Plain Old Fact Science
E. If You Know Anyone Who Is Afraid Of Gay People, Here's A Cartoon That Will Ease Them Back To Reality

E

Demonstrations are sample inputs and outputs that illustrate the desired behavior or response. This method does not change the underlying parameters of the GPT models; rather, it leverages the model's ability to recognize patterns and apply them to new, similar tasks by conditioning on input-output examples. Previous studies have shown that in-context learning can significantly boost a model's accuracy and relevance in responses compared with zero-shot prompting. Xie and Min (2022) explain this using a Bayesian inference framework, suggesting that in-context learning involves inferring latent concepts from pretraining data, utilizing all components of the prompt (inputs, outputs, formatting) to locate these concepts, even when training examples have random outputs. Xie et al. (2021) further elucidate that in-context learning can emerge from models pretrained on documents with long-range coherence, requiring the inference of document-level concepts to generate coherent text. They provide empirical evidence and theoretical proofs showing that both transformers and long short-term memories (LSTMs) can exhibit in-context learning in synthetic data sets.

We design in-context learning prompts as shown in Figure 2. Our prompts include demonstrations of best-

Figure 2. (Color online) In-Context Learning Prompt for Headline Selection



You are an expert in digital media marketing and content strategy, specializing in optimizing headlines for maximum click-through rates (CTR) and increasing ad revenue. Your deep understanding of reader behavior, SEO, and digital engagement allows you to evaluate headlines for their potential to capture attention and drive clicks.

I have a news article, and I need to choose the catchiest headline from the following list. By "catchiest," I mean the headline that is most likely to generate the highest CTR, thereby increasing ad revenue for the news company. The selected headline should:

1. Capture readers' attention immediately.
2. Appeal to emotions, curiosity, or urgency.
3. Be engaging enough to make readers want to click on the article.
4. Follow best practices for digital news, considering SEO, shareability, and intrigue.

I will also provide examples of multiple headline sets that have performed well in the past, along with the best-performing headline index for each set to guide your selection.

Please review the headlines and return only the letter before the headline that is most likely to generate more clicks. **No explanation is needed. No need to return the headline, only the letter.**

Here are examples of headlines that have worked well before:
Example 1:
A. New York's Last Chance To Preserve Its Water Supply
B. How YOU Can Help New York Stay Un-Fracked In Under 5 Minutes
C. Why Yoko Ono Is The Only Thing Standing Between New York And Catastrophic Gas Fracking

Best-performing headline: C

Example 2:
[Second demonstration]

Here are the headlines I need you to evaluate:

A. Hey Dude. If You Have An Older Brother, There's A Bigger Chance You're Gay.
B. Here's The Science, Here's The Gay. Open Your Brain, They Were Born That Way!

C. I've Got Some News For You. Being Gay Is Genetic. Being Irrationally Afraid Of Gay, Not So Much.
D. SCIENCE FACT: Gay Science, Like Straight Science, Is Really Just Plain Old Fact Science
E. If You Know Anyone Who Is Afraid Of Gay People, Here's A Cartoon That Will Ease Them Back To Reality

E

performing headlines and ask the model to evaluate the subsequent headlines. The demonstrations are selected from the training set, and for different prompt inquiries, we use the same demonstration to maintain comparability. A few demonstrations are typically sufficient to guide the model. Improvement tends to have diminishing returns quickly; that is, adding more demonstrations does not significantly enhance performance after a certain point (Min et al. 2022). In our numerical experiments, we use two or five demonstrations to investigate the impact of the number of examples on performance.[7]

An interesting aspect of in-context learning is the model's robustness to incorrect labels in the demonstrations. Min et al. (2022) show that providing randomly selected labels as the answers in demonstrations barely affects performance, and they highlight other critical aspects such as the label space, input text distribution, and sequence format. This is likely because the model relies more on the distribution of input text, label space, and format rather than specific input-label mappings. That is, in-context learning seems to benefit from recognizing general patterns and structures in the data, which the model infers from its pretraining rather than the actual labels. To examine if this is the case in our setting as well, we also consider experiments where we intentionally provide incorrect labels in the demonstrations.

3.1.3. Performance Analysis of Prompt-Based Approaches.

Before presenting the results, we first make a note that we do not find any significant difference between GPT-4 and GPT-4o; therefore, we only report the results from GPT-3.5 and GPT-4 for simplicity. The results of all the prompt-based experiments are shown in Table 4. A more detailed pairwise comparison and t-test analysis of all the different prompts are available in Web Appendix Section B.1. The second column of Table 4, n_{demo}, represents the number of demonstrations used for in-context learning, with $n_{demo} = 0$ indicating a zero-shot prompting. Additionally, for all the in-context learning experiments, we consider two scenarios: one where the demonstrations are shown with ground truth labels and another where they are shown with incorrect labels. The third column, "Incorrect Label," indicates whether the labels were correct (incorrect label = 0) or incorrect (incorrect label = 1).

Table 4. Results for Prompt-Based Approaches with Different GPT models, the Number of Demonstrations, and Whether We Intentionally Provide Incorrect Solutions in the Demonstration

Model	n_{demo}	Incorrect label	Accuracy in the test set (%)
Random guess	—	—	33.02
GPT-3.5	0	—	29.76
GPT-3.5	2	0	34.23
GPT-3.5	2	1	31.60
GPT-3.5	5	0	30.95
GPT-3.5	5	1	30.59
GPT-4	0	—	37.85
GPT-4	2	0	**39.96**
GPT-4	2	1	38.77
GPT-4	5	0	39.17
GPT-4	5	1	38.28

Notes. Accuracy is defined as the ratio of correct answers over the test set size. The best performance in all test data is boldfaced.

We find that prompt-based approaches using GPT-3.5 are not significantly better than random guess or are even slightly worse. In comparison, experiments with GPT-4 have accuracy rates around 38%–40%, which is a significant improvement in performance compared with GPT-3.5, and all settings for GPT-4 outperform the random guess. For GPT-4, in-context learning with two demonstrations slightly improves performance over zero-shot prompting in GPT-4, though increasing the number of demonstrations from two to five does not result in a significant difference in performance. The difference between using correct labels and incorrect labels in the demonstrations is also not significant for in-context learning. This confirms that in-context learning helps GPT understand the task better through the input-output examples, but it fails to leverage the information in mapping from inputs to outputs effectively.

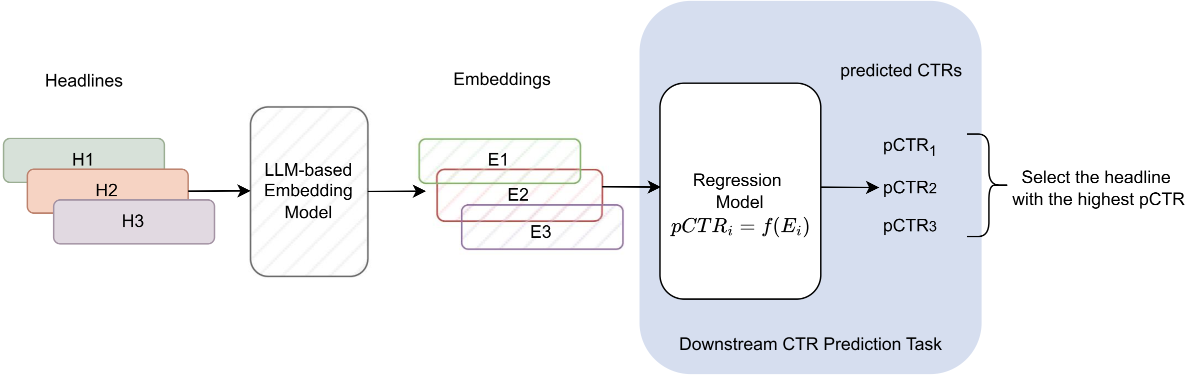
In summary, we find that prompt-based methods have low accuracy (no more than 40%), slow running time (compared with other methods introduced later in this section), and high monetary cost (see Web Appendix Section E for a detailed discussion of costs and running time). All of these drawbacks make them unsuitable for real-time business applications; that is, these methods cannot replace experimentation for content selection in digital platforms.

3.2. CTR Prediction Models with LLM-Based Text Embeddings

We now consider another LLM-based method for the headline selection task. Specifically, we leverage the text embeddings from OpenAI and utilize regression models, including linear regression (linear) and multilayer perceptron (MLP), to predict the CTRs of headlines for the same article and then select the one with the highest predicted CTR as the winning headline.

Text-embedding techniques transform large chunks of text, such as sentences, paragraphs, or documents, into numerical vectors. These embedding vectors can then be used in a variety of applications, including text classification, where texts are grouped into categories; semantic search, which improves search accuracy by understanding the meaning behind the search queries; and sentiment analysis, which is used to determine the

Figure 3. (Color online) The Pipeline of the Headline Selection Using LLM Text Embeddings



Notes. We use an A/B test with three headlines for illustration. Headlines 1, 2, and 3 are natural language sentences, whereas embeddings 1, 2, and 3 are numerical vectors.

emotional tone of a piece of text.[8] See Patil et al. (2023) for a more detailed survey of the most recent text-embedding models and their applications.

An interesting question is why text embeddings obtained from LLM models trained for the next word/token prediction task with the cross-entropy loss are able to produce informative features for downstream tasks. Saunshi et al. (2020) provide an explanation by demonstrating that typical downstream tasks can be reformulated as sentence completion tasks, which can be further solved linearly using the conditional distribution over words following an input text. Thus, the embeddings, which can be roughly viewed as the next token probability generated from a next-word prediction task, inherently capture the contextual information and relationships between words, making them useful for various downstream tasks such as sentiment analysis and headline selection (as our experiments illustrate).

Figure 3 illustrates the conceptual framework of our approach. We first use the LLM-based embedding model to transform the original headlines into embeddings. Then, we use the embeddings as inputs and CTRs as outputs to train the downstream CTR prediction model. For the test set, we leverage the trained regression model to predict the CTR of one headline at a time. After getting all CTR predictions of headlines in an A/B test, we rank the headline with the highest CTR as the winning headline. We consider four embedding models, including the OpenAI embedding model, Llama-3-8b, Word2Vec, and BERT. Note that we do not train the embedding model and only train the downstream CTR prediction model using the training data set, which contains 12,376 headline A/B tests. A detailed introduction to embedding models and the implementation specifics is provided in Web Appendix Section C.1.

3.2.1. Performance Analysis of Embedding-Based Approaches.

In Table 5, we present the performance of CTR prediction models based on OpenAI's embeddings, including the 256-dimensional embedding (OpenAI-256E) and the 3,072-dimensional embedding (OpenAI-3072E, which is the maximum size). The full performance results, including Llama-3-8b, Word2Vec, and BERT, can be found in Web Appendix Section C.2. At a high level, we see that all the embedding-based regression models obtain better accuracy than prompt-based approaches. This performance can be attributed to the fact that these models are trained on a customized data set, allowing them to learn decision rules tailored to the headline selection task. Because these models can leverage a larger labeled data set during training, they are more generalizable and robust. In contrast, in-context learning relies on the LLM's ability to understand and generalize from only a few demonstrations. Furthermore, the task of choosing the best headline from multiple candidates may not be a common scenario in OpenAI's training corpus. Consequently, GPT models may struggle to generalize effectively from the limited examples provided in prompts, leading to less accurate predictions.

We also find that for OpenAI, higher-dimensional embeddings do not lead to higher prediction accuracy. Given that the 256-dimensional embedding stores only

Table 5. The Performance of OpenAI Embeddings and CTR Prediction Models in the Test Data

Embedding model	CTR prediction	Accuracy in the test data (%)
OpenAI-256E	Linear	**46.28**
	MLP	44.38
OpenAI-3072E	Linear	43.06
	MLP	45.17

Notes. All models use the same training data, and accuracy is evaluated in the test data. Note that the linear model's training is a deterministic process after fixing the data, whereas MLP training incurs randomness via stochastic gradient descents. Therefore, we rerun MLP training three times with different random seeds and report the average accuracy from three reruns. The best performance in all test data is boldfaced.

essential information, whereas the 3,072-dimensional embedding also captures finer details, this performance gap is possibly because of the overfitting with the limited size of the training set. Predictions using higher-dimensional embeddings may cause the prediction model to focus too much on details and overfit the training set. A more striking observation is that increasing the complexity of the CTR prediction model does not necessarily improve predictions for OpenAI embeddings. We hypothesize that with more dimensions in text embeddings, the CTR prediction task requires a more sophisticated model because of the higher risk of overfitting. Overall, our findings suggest that a linear CTR prediction model on top of fixed text embeddings from OpenAI can effectively capture the attractiveness of content.

We now investigate three other aspects of these models and summarize our findings below. For brevity, we simply discuss the findings here and refer interested readers to the Web Appendix for details.

- We compare performance under different embedding models. We observe that OpenAI embeddings outperform Llama-3-8b, Word2Vec, and BERT-based models. See Web Appendix Section C.2 for details.
- We quantify the impact of training sample size on predictive accuracy. In Web Appendix Section C.3, we train the model using varying proportions of the training data and evaluate its performance. We find that larger training data sets lead to higher accuracy rates, although the improvement is gradual. For instance, increasing the training size from 20% to 70% of the total data set raises accuracy from 44% to 46% on the test data.
- There are two potential threats to the inference from our analysis. First, one may be concerned as to whether Upworthy's data has been used as part of OpenAI's training corpus, which could potentially invalidate the results. A second concern is that readers' preferences changed over time and that the current sample splitting method fails to take into account when the headline was created. We examine both of these issues and provide a detailed discussion and robustness checks to alleviate concerns around them in Web Appendix Section C.4.

3.3. Fine-Tuning Open-Source LLMs with LoRA

In Section 3.2, we trained prediction models using text embeddings as fixed inputs, bypassing the need to train the LLM itself. The best-performing embedding-based model reaches an accuracy of 46.28% on the test set without modifying any parameters in the LLM. This raises a natural question: if we can train LLMs directly, can we achieve better performance? However, training LLMs from scratch is costly and resource intensive. Therefore, in this section, we turn to fine-tuning techniques, which involve partially adjusting the parameters in the LLM to improve performance while keeping the computational cost manageable. Note that to fine-tune LLMs, we need to be able to access the underlying model parameters. This is only feasible for open-source LLMs such as BERT and Llama.[9] The goal of fine-tuning is to adapt a general-purpose LLM for specific downstream tasks through a process typically involving supervised learning using labeled data sets.

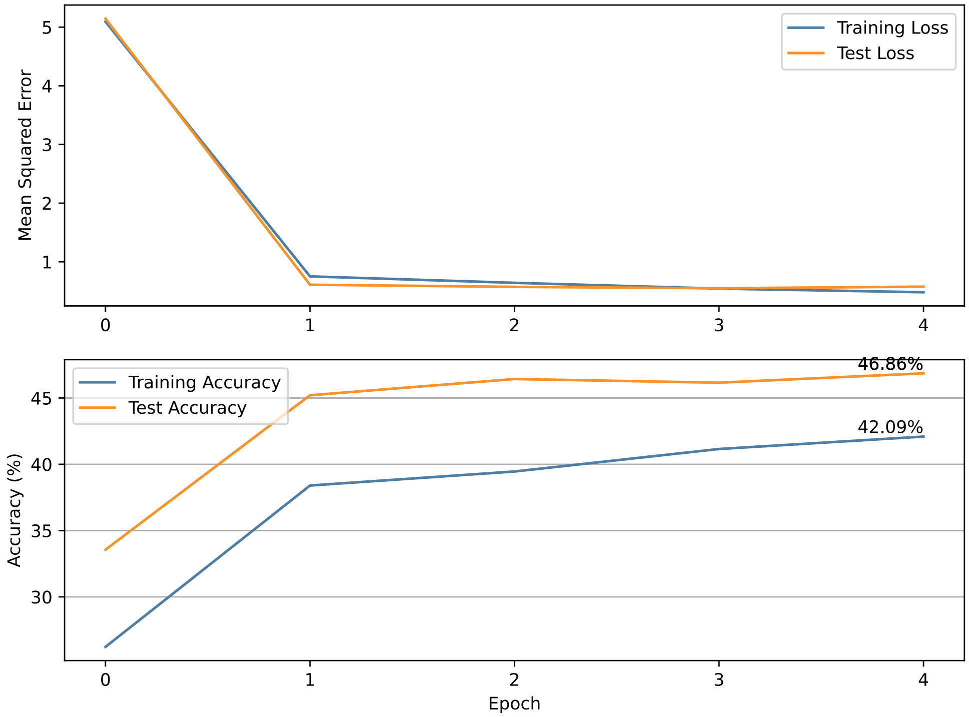
For the fine-tuning task, we use Meta's Llama-3-8b as our primary base model. Zhao et al. (2024) consider a large-scale experiment where they fine-tune different open-source LLMs and show that for NLP tasks, including news headline generation, Llama-3 offers one of the best performances. Llama-3 offers two versions: one with eight billion parameters (Llama-3-8b) and another with 70 billion parameters (Llama-3-70b). We opted for the smaller version because it is easier to fine-tune, and as we will see, even with this smaller model, we see evidence of overfitting (see Section 3.3.1 for details). Hence, we avoid fine-tuning the 70-billion-parameter model, where overfitting problems are likely to be exacerbated. Background information related to the transformer block, the fundamental component of most LLMs, including Llama-3, and the basic architecture of Llama-3 is given in Web Appendix Section D.1.

Traditional fine-tuning of LLMs involves updating a substantial number of parameters, which can be computationally expensive and memory intensive. To efficiently fine-tune the Llama-3 model on hardware with limited GPU memory, we employ LoRA, a popular parameter-efficient fine-tuning (PEFT) method developed by Hu et al. (2022), enabling practical fine-tuning of large-scale LLMs on medium-scale hardware.[10] LoRA operates under the assumption that updates during model adaptation exhibit an intrinsic low-rank property and introduce a set of low-rank trainable matrices into each layer of the transformer model. This approach leverages the low-rank decomposition, which significantly reduces both memory usage and computational time through a significantly reduced number of trainable parameters. More details about LoRA fine-tuning and the implementation are given in Web Appendix Section D.2 and Section D.3, respectively.

3.3.1. Performance Analysis of Fine-Tuning-Based Approaches.

We report the loss curve and accuracy curve during the LoRA fine-tuning process in Figure 4. We observe an accuracy of 46.86% on the test data, which is marginally better than the best performance of 46.28% using OpenAI embeddings in Section 3.2. However, such similar performance should not undermine the utility of fine-tuning. Comparing the fine-tuning of Llama-3-8b with the use of OpenAI embeddings is not straightforward because of inherent differences in the

Figure 4. (Color online) Loss Curve and Accuracy Curve as the Number of Training Epochs Increases



models' capabilities. To highlight the benefits of fine-tuning, we observe that the Linear CTR prediction model using embeddings derived from Llama-3-8b (42.78% on the test data, reported in Web Appendix Section C.2, Table A2) performs worse than the LoRA fine-tuned Llama-3-8b. This demonstrates that fine-tuning Llama-3-8b can enhance performance compared with using Llama-3-8b embeddings alone.

In Figure 4, we observe that the accuracy on the test set (46.86%) is higher than on the training set (42.09%), which is uncommon in machine learning tasks. The reason for this discrepancy might be that the split of the training and test data sets results in some easily distinguishable headlines being included in the test set, making the training set more challenging than the test set. To verify our hypothesis, in Web Appendix Section D.4, we try several additional random splits of the training and test data sets and evaluate their performance on them. We find that in these other splits, the test performance is slightly worse than the training performance.

3.4. Summary of Performance of Pure LLM–Based Methods

We summarize the performance of various pure-LLM-based methods in Table 6. We also report the results of human respondents on a similar task but with paired headline comparison. Details of the survey questions and the accuracy analysis of human respondents are shown in Web Appendix Section A.

There are four main takeaways from these comparisons. First, human respondents cannot easily predict which headline would be more engaging and likely to lead to higher CTRs. This is consistent with the fact that media firms (and even expert editors) rely on experimentation to decide which headlines and articles to display to users. Second, whereas simple prompt-based approaches, such as in-context learning with

Table 6. Summarization of Performance of Various Pure LLM–Based Methods

Method	Accuracy in test set (%)	Notes
Random guess	33.02	
Human respondents	—	Similar to random guess as tested on 4,571 human responses
Prompt	39.96	GPT-4 with in-context learning
Embedding	46.28	OpenAI-256E embedding model with linear regression
Fine-tuning	46.86	LoRA fine-tuned Llama-3-8b

Notes. All results in this table have been reported before. We only report the best performance of each method.

examples, are better than purely random guesses or human predictions, they nevertheless do not provide sufficiently high accuracy. Third, both embedding-based models and fine-tuning perform equally well (at least in our setting) and provide equally good performance, with around 46%–47% accuracy on the test set. They each have their pros and cons. The main advantage of the linear CTR prediction model with OpenAI embeddings is that it is easy to implement and use in business settings because it does not require any significant model training/deployment. However, embeddings from the OpenAI models are proprietary, and as such, we lack visibility into the exact nature of the embeddings. In contrast, fine-tuning approaches use open-source models such as Llama-3-8b, and we have access to the underlying parameters of the LLM. However, this transparency comes with costs: fine-tuning LLMs requires nontrivial training and deployment skills that not all businesses may have or want to invest in. The fourth and most important takeaway is that whereas pure-LLM-based approaches are informative, they are unable to reach the accuracy of experimental approaches. That is, firms cannot replace content experimentation with LLM-based predictions.

An important caveat here is consumer preferences may have changed significantly because the time period of the experiment (2013–2015). As such, the prediction numbers shown here may not represent LLMs' ability to predict the preferences of consumers today.[11] As such, the performance of LLM-based methods in Section 3 should be mainly used for relative comparisons with each other. More broadly, both LLM capabilities and consumer preferences will continue to evolve. Therefore, if a firm/manager is concerned about the generalizability/accuracy of a specific LLM-based prediction model in their current business environment, the simplest solution is to retrain their LLM-based models more frequently and collect up-to-date labeled data to maintain relevance.

Finally, we discuss the monetary costs associated with three LLM-based approaches. The costs of prompt-based, embedding-based, and LoRA fine-tuning in our analysis are around $15.87, $0.29, and $3, respectively. The cost of prompt-based and embedding-based comes from access to OpenAI's API in order to get responses and text embeddings. The cost of fine-tuning comes from renting a cloud computing server equipped with one NVIDIA A100 GPU. Overall, we find that the prompt-based approaches are the most expensive, whereas embedding and fine-tuning are the cheapest in terms of cost. However, in terms of run time and effort, fine-tuning is more costly. The detailed comparisons of the cost, runtime, and engineering effort required for each pure-LLM method are discussed in detail in Web Appendix Section E.

4. Our Approach: LLM-Assisted Online Learning Algorithm

So far, we have seen that pure-LLM-based approaches can help identify the best headline with 46.86% accuracy using the LoRA fine-tuning approach. The main advantage of pure LLM–based approaches is that they can be deployed without any experimentation; that is, the manager can simply use the LLM-based model to predict which headline is the best and go with it. Thus, there is no need to use any traffic for experimentation. The downside, of course, is that these models are not fully accurate; in some cases, they are likely to mispredict which headline is the best. This can, of course, lead to lower clicks and revenues for the firm. Thus, it is not clear that relying on pure-LLM-based methods will lead to better overall outcomes for the firm.

On the other hand, we have the current status quo, which is the use of experimentation-based approaches to choose headlines/content. As mentioned in Section 1, there are two broad experimentation-based methods: (1) A/B tests, and (2) online learning algorithms or bandits. In particular, the standard practice at Upworthy during the time of the data collection was A/B tests, where the firm allocated a fixed amount to different headlines and then chose the best-performing one for the remaining traffic. This approach is also known as explore and commit and is commonly used in the industry because of its simplicity and low engineering cost; see chapter 6 of Lattimore and Szepesvári (2020). The main benefit of this approach is that with a sufficiently large amount of traffic allocated to the A/B test, the resulting inference is unbiased; as such, it is the gold-standard approach for identifying the best arm (or headline in this case). The downside, of course, is that by assigning traffic to all the headlines during the exploration phase or A/B test, the firm incurs high regret (i.e., low profits because some headlines in the test are likely to have low CTRs). Therefore, many modern technology and media firms such as the *New York Times* and Yahoo use online learning algorithms or bandits that leverage the explore-exploit paradigm to lower the experimentation cost while still learning (Lattimore and Szepesvári 2020). These algorithms move traffic away from poorly performing arms/headlines dynamically in each period, based on the observed performance of each arm till that period. Because of this greedy behavior, they tend to switch to the best-performing arms quickly and incur lower regret. Nevertheless, one drawback of these methods is that they start with the assumption that all headlines are equally good; that is, they suffer from a cold-start problem, which can result in wasting traffic on the weaker headlines early on.

In this section, we present a framework that combines the benefits of LLM-based models with experimentation-based methods. Our key idea is that firms can address the cold-start problem in content experiments by leveraging predictions from LLM-based approaches as CTR priors in the first step and combining them with an online learning algorithm in the second step. Thus, our approach, termed LLM-assisted online learning algorithm, continuously optimizes traffic allocation based on LLM predictions together with the data collected from the online experiment in real time.

The rest of this section is organized as follows. In Section 4.1, we present the media firm's problem and the standard bandit framework, and then in Section 4.2, we present our LOLA framework. Next, in Section 4.3, we discuss the implementation of our proposed algorithm, benchmarks, and the experimental setup. We present our numerical results in Section 4.4, and finally, in Section 4.5, we show how our LOLA approach can be easily generalized to a wide variety of settings and present a set of natural variants and extensions of the basic LOLA framework.

4.1. Bandit Framework

Multiarmed bandits (MAB) is a class of sequential decision-making problems where a learner must choose between multiple arms over periods to maximize cumulative reward. Whereas there is a large literature on it in the machine learning and statistics community, the literature in marketing is small but growing; researchers have used it for optimizing website design, pricing, advertising, and to learn consumer preferences (Hauser et al. 2009, Schwartz et al. 2017, Misra et al. 2019, Liberali and Ferecatu 2022, Aramayo et al. 2023, Jain et al. 2024). In our setting, different arms refer to different headlines associated with the same article, and periods refer to impressions where different headlines with the same article are displayed. Let $[K] := \{1, 2, \ldots, K\}$ represent the set of K arms and $[T] := \{1, 2, \ldots, T\}$ represent the entire decision horizon for a test. At each time step t, the learner selects an arm $a_t \in [K]$ and receives a reward r_{a_t} drawn from a probability distribution specific to the chosen arm a_t. We assume each arm has a constant CTR, and the reward is the click feedback, which follows the Bernoulli distribution with a probability equal to the CTR.

The learner's goal is to minimize cumulative regret, which is defined as the difference between the reward obtained by always choosing the optimal arm and the reward actually accumulated by the learner under the policy played by the learner. Mathematically, the regret R_T after T trials is given by

$$R_T = \max_{k \in [K]} \sum_{t=1}^{T} r_{a^*} - \sum_{t=1}^{T} r_{a_t}, \tag{1}$$

where $a^* = \arg\max_{k \in [K]} \mathbb{E}[r_k]$ is the optimal arm with the highest expected reward.

In MAB settings with finite arms, the standard algorithm that is typically used is the UCB algorithm (Auer et al. 2002). UCB stands out among all online learning algorithms because it is provably asymptotically sublinear; that is, $R_T/T \to 0$ as $T \to \infty$. See chapter 7 of Lattimore and Szepesvári (2020) for the formal proof. The UCB policy is straightforward: it selects the arm that maximizes the upper confidence bound on the estimated rewards. Specifically,

$$a_t = \arg\max_{k \in [K]} \left(\bar{\mu}_k^t + \sqrt{\frac{2 \log(1/\delta)}{n_k^t}} \right), \tag{2}$$

where $\bar{\mu}_k^t$ is the estimated mean reward of arm k up to time t, n_k^t is the number of times arm k has been played up to time t, and δ is the confidence level, which quantifies the degree of certainty. The intuition behind UCB's effectiveness in balancing exploration and exploitation lies in its construction. By adding a confidence term $\sqrt{2 \log(1/\delta)/n_k^t}$ to the estimated mean reward $\bar{\mu}_k$, UCB ensures that arms with fewer selections (higher uncertainty) are given a chance to be explored. As time progresses and more data are collected, the confidence term diminishes, allowing the algorithm to exploit arms with higher observed rewards. This dynamic adjustment enables the UCB algorithm to explore underexplored arms while exploiting arms with high rewards systematically, thus effectively addressing the exploration-exploitation trade-off inherent in the MAB problem. However, we need to set the key hyperparameter δ such that it is small to ensure optimality with high probability but not so small that suboptimal arms are overexplored; see chapter 7.1 of Lattimore and Szepesvári (2020) for a detailed discussion. Typically, we set the confidence term as $\alpha\sqrt{\log t/n_k^t}$ and fine-tune the scale term α for specific problems.

4.2. LOLA

We now discuss our approach LOLA and how the standard UCB approach discussed above can be extended to include information from LLMs in the first step preceding active experimentation. LOLA has two key steps, and we describe each in detail.

In the first step, we use LLMs to get an initial prediction of each headline's CTR. Note that the LLM training phase in this algorithm description considers only one prediction model without an explicit model selection process, as our focus is on online learning and the integration aspect. However, the LOLA algorithm is agnostic to the choice of LLM prediction models and can easily incorporate model selection if multiple models are available. Specifically, multiple prediction models can be trained during the training phase, and in the hyperparameter fine-tuning phase, model performance can be compared on a validation subset before proceeding with the fine-tuning of n^{aux} and α. In this way, model selection can be treated as an additional hyperparameter within the LOLA framework. Based on numerical results in Section 3 (the evaluation of which is based on 20% test data) and the model selection results in Web Appendix Section F (the evaluation of which is based on 10% validation data), we choose the consistent best-performing approach, LoRA fine-tuned Llama-3-8b model as the best CTR prediction model in the first step of the algorithm.

In the second step, we build on the bandit framework from Section 4.1 to design an online algorithm that incorporates the LLM-based CTR predictions from the first step. Inspired by the 2-UCBs policy designed by Gur and Momeni (2022), we propose Algorithm 1, titled LLM-Assisted 2-Upper Confidence Bounds (LLM-2UCBs). Essentially, in this algorithm, we treat the LLM's CTR predictions as auxiliary information prior to the start of online experiments, equating this auxiliary CTR information to additional impressions and click outcomes for each arm.

Algorithm 1 (LLM-Assisted 2-Upper Confidence Bounds)

```
1. **LLM training phase:** Train an LLM-based prediction model $\mathcal{M}(x)$ for CTRs using historical data samples, where $x$ is the contextual information for arms.
2. **Hyperparameter fine-tuning:** Use another subset of data to select two hyperparameters for the algorithm, $\alpha \in \mathbb{R}^+$ for controlling the upper bound, $n^{\text{aux}}$ as the LLM's equivalent auxiliary sample size.
3. **Online learning phase:** Initialize the number of periods $T$, number of arms $K$, LLM-based CTR prediction $\bar{\mu}_k^{\text{aux}} \leftarrow \mathcal{M}(x_k)$, regular CTR initialization $\bar{\mu}_k^1 = 0$, accumulated impressions $n_k^1 \leftarrow 1$, and accumulated clicks $c_k^1 \leftarrow 0$ for all arms $k \in [K]$.
4. **for** $t = 1$ **to** $T$ **do**
5.     Calculate the first UCB for all arms $k \in [K]$: $U_k^1 = \bar{\mu}_k^t + \alpha\sqrt{\log t / n_k^t}$.
6.     Calculate the second UCB for all arms $k \in [K]$:
           $U_k^2 = \frac{c_k^t + \bar{\mu}_k^{\text{aux}} n^{\text{aux}}}{n_k^t + n^{\text{aux}}} + \alpha\sqrt{\frac{\log t}{n_k^t + n^{\text{aux}}}}$.
7.     Play the arm $a_t = \arg\max_{k \in [K]} \min\{U_k^1, U_k^2\}$.
8.     Observe the payoff $r_{a_t}$
9.     Update $n_{a_t}^{t+1} \leftarrow n_{a_t}^t + 1$, $c_{a_t}^{t+1} \leftarrow c_{a_t}^t + r_{a_t}$, and $\bar{\mu}_{a_t}^{t+1} \leftarrow c_{a_t}^{t+1} / n_{a_t}^{t+1}$.
10. **end for**
```

There are two hyperparameters in LLM-2UCBs: (1) the upper bound control hyperparameter, α, that also shows up in the regular UCB algorithm, and (2) the hyperparameter n^{aux}, which is specific to our proposed algorithm and indicates the equivalent auxiliary samples for initialization. At a high level, we can regard the LLM predictions as prior information or the information flow before the algorithm starts. This prediction is approximately equivalent to initially generating n^{aux} impressions for each arm and observing their outcomes to obtain the sample average CTR, $\bar{\mu}_k^{\text{aux}}$. The hyperparameter fine-tuning process is simple. We can utilize another randomly sampled data set to test combinations of two hyperparameters and select the combination that yields the highest accumulated rewards.

During the online learning phase, the key step is balancing the exploration and exploitation trade-off. We capture this trade-off using the 2-UCBs rule. Essentially, we have two valid UCBs: U_k^1 is the regular UCB, which uses observed impressions and clicks to construct the mean reward estimator and adds a regular upper bound. The second UCB, U_k^2, incorporates LLM predictions. The calculation of U_k^2 is straightforward. The mean reward component is simply the sample average, taking into account the "auxiliary" samples generated by the LLM before the learning phase begins. These auxiliary samples are imaginary and approximate the richness of information contained in the LLM prediction. The next step is to choose the smaller of the two UCBs, $\min\{U_k^1, U_k^2\}$, as the final UCB for selecting the arm. Note that if $n^{\text{aux}} = 0$, the 2-UCBs rule is the same as the standard UCB rule. Conversely, when $n^{\text{aux}} \to \infty$, the 2-UCBs rule is equivalent to a pure exploitation policy (a.k.a. greedy policy) using the LLM-based CTR prediction. Thus, when n^{aux} is a positive integer, the 2-UCBs algorithm incorporates information from both LLMs and the actual click outcomes. After playing the arm and observing the outcome, we update the impressions, clicks, and mean reward accordingly, following standard bandit algorithm procedures.

The intuition behind using two UCBs instead of a single UCB incorporating LLM predictions U_k^2 is as follows: if we rely solely on U_k^2, and the LLM's CTR predictions are significantly incorrect, that is, overestimating the CTR of a poorly performing headline (underestimating the CTR of the best headline is of less risk because there are more suboptimal headlines than the one and only best headline), it would take many rounds to correct this error, especially when the number of auxiliary samples, n^{aux}, is large. On the other hand, the first UCB U_k^1, though unbiased, suffers from large stochastic noise, requiring large n_k^t to get an accurate estimator. By choosing the minimum of the two UCBs, we balance these opposing risks effectively. However, the formalism for this approach is only partially addressed in prior literature (Gur and Momeni 2022), which operates under a different setting without LLMs. In particular, Gur and Momeni (2022) show 2-UCBs is optimal under the assumption that the auxiliary data are generated from a linear function class, including the standard UCB setting with exogenous auxiliary samples. The formalism in our context would require additional assumptions about the errors in the LLM's predictions, which remains an open research question with limited understanding. We leave this formalization for future work.

The LOLA algorithm offers three advantages. First, it is easy to implement, requiring the tuning of only two hyperparameters. Second, it can be used in conjunction with any LLM-based CTR prediction model in the first step and any outcome of interest in the second step. We use clicks as the outcome of interest given our application setting and data; however, other outcomes, such as time spent on a page or other measures of engagement, can be easily used instead of clicks. Finally, it is intuitive and flexible. Because it can easily accommodate the edge cases, for example, one where the manager has no auxiliary information or one where the manager wants to fully rely on the LLM predictions, there is no need for extra engineering effort to set up separate systems for different edge cases. This adaptability is valuable given the rapid advances in LLM technology. When the capability of the pure-LLM method continues to advance, the manager can easily fine-tune and increase n^{aux} to take more advantage of LLM.

As a final note, our algorithm does not update the LLM during the online learning phase. Instead, we use the LLM fine-tuned on historical data solely for initialization purposes. That being said, the algorithm can be readily modified to fine-tune LLMs with online data collected during the learning phase. Whereas such fine-tuning could potentially improve LOLA's performance, the improvement is expected to be limited in our current data set. As shown in Figure A3 in Appendix Section C.3, achieving a significant performance boost in LLMs would require a much larger data set than is currently available to us.

4.2.1. Discussion.

We now provide a broader discussion on LOLA. We note that the general idea that priors can impact regret and rewards is not new and has existed for some time; see Bubeck and Liu (2013) and Russo and Van Roy (2014). These earlier papers primarily focus on theoretical analyses and often rely on assumptions about the quality of priors. However, because we do not have a theoretical understanding of LLM predictions, it is unclear how these predictions can be translated to priors or the nature of such priors. Separately, the 2-UCBs algorithm by Gur and Momeni (2022) was designed for a setting where the experimenter has access to auxiliary data. However, their algorithm assumes that the researcher has access to both the size of this auxiliary sample (n^{aux}) and knowledge of the prediction error in this sample. Thus, neither of these earlier approaches is directly applicable to our setting.

A key challenge in our setting is that the LLM predictions cannot be naively treated as priors because we do not know their prediction error or theoretical properties. The main novelty of LOLA comes from the idea that we can treat the LLM predictions as coming from a "pseudosample" and that we can fine-tune the size of this pseudosample (n^{aux}) for a given application based on prior data. Thus, we build on the 2UCBs algorithm but modify it to accommodate LLM predictions that do not come from a real sample and whose quality and theoretical properties are unknown.

4.3. Benchmarks and Implementation Details

We now discuss the implementation details and present a set of benchmark algorithms. First, we randomly split our data into three exclusive folds based on the test ID, ensuring that all headlines in the same test are in the same fold and that the same headline does not appear in different folds. We use 70% of the data for training, 10% for hyperparameter fine-tuning, and 20% for testing. The training data are used to train the LLM models and are only used by our proposed LOLA approach (the LLM-2UCBs specifically) and pure LLM–based benchmarks. All hyperparameters in all the algorithms (both LOLA and the benchmark algorithms) are selected using the fine-tuning data, and the performance of the algorithms is evaluated on the test data.

We now describe a set of commonly used algorithms that serve as benchmarks and our LOLA approach.

- Explore and commit: In this approach, the firm runs the A/B test for a fixed number of periods (i.e., explores) and then commits to the best-performing arm at that stage; see chapter 6 of Lattimore and Szepesvári (2020). This is the standard A/B test approach used by many digital firms in the industry. In particular, Upworthy's approach during the data collection period closely resembles the E&C algorithm—recall that they assign traffic to candidate headlines with equal probability during the experiment phase and then show the winning headline to all future impressions. The E&C method has a single hyperparameter that needs tuning: the proportion of periods for exploration. We consider the following proportions: {0.1, 0.2, 0.3, 0.4, 0.5}, and we choose the one with the highest accumulated rewards based on the 10% of data used for hyperparameter tuning. We finally choose 0.2 for exploration.
- Pure LLM–based approach: In this approach, the firm uses LLMs to predict the CTR for all the arms (headlines) and then allocates all the traffic to the arm with the highest predicted CTR. This is equivalent to a fully greedy policy based on LLM predictions. We use the training data to learn a model that predicts the CTR of a headline based on an LLM (either through text embeddings or LoRA fine-tuning). In our numerical

study, we use the best-performing pure LLM–based approach based on our experiments, that is, the LoRA fine-tuned Llama-3-8b, which gives a 46.86% accuracy.[12]

- Standard UCB: This is the standard UCB algorithm discussed in Section 4.1. Here, the firm does not use any first-stage LLM predictions as an input, and there is only one hyperparameter to fine-tune, the confidence control, α. We consider values from the set $\{0.02, 0.04, \ldots, 0.10\}$ and find that $\alpha = 0.08$ gives the highest accumulated rewards.
- LOLA: This is our proposed approach. In the first step, we use CTR predictions from the best-performing pure LLM–based CTR prediction model, that is, the LoRA fine-tuning Llama-3-8b with mean squared error (MSE) loss. Note that this is the same model that we use for the pure LLM–based approach, which allows for a fair comparison of the two approaches. For the second step, we use the LLM-2UCBs algorithm as outlined in Section 4.2. We fine-tune the combination of the two hyperparameters on the fine-tuning data, and we consider $n^{\text{aux}} \in \{600, 800, 1,000, 1,200, 1,400\}$ and $\alpha \in \{0.02, 0.04, \ldots, 0.10\}$. Based on fine-tuning, we choose $n^{\text{aux}} = 1,000$ and $\alpha = 0.08$ for our setting. We refer interested readers to Web Appendix Section G for a detailed sensitivity analysis of n^{aux}.

Note that the first two methods can be viewed as special cases of our proposed LOLA algorithm. Both our algorithm and the pure LLM–based approach use the exact same CTR prediction algorithm. Hence, the pure LLM–based approach (a.k.a. pure exploitation, greedy algorithm) is a special case of LOLA wherein we set $n^{\text{aux}} = \infty$, indicating complete trust in the LLM prediction results. This method fully relies on the LLM CTR prediction, directly selecting the headline with the highest predicted CTR throughout the entire horizon. Similarly, the standard UCB algorithm is a special case of LOLA with $n^{\text{aux}} = 0$, indicating no reliance on LLM CTR predictions. Again, note that we use the same hyperparameter $\alpha = 0.08$ across both algorithms, which ensures that our algorithm defaults to the standard UCB with $\alpha = 0.08$ when we ignore the CTR predictions from the LLM.

- UCB with LLM priors: To demonstrate the effectiveness of the 2-UCBs policy $\min\{U^1, U^2\}$ compared with using only the second UCB U^2, we also run the UCB with LLM as priors, that is, play the arm $a_t = \arg\max_{k \in [K]} U_k^2$. This comparison highlights how selecting the minimum of the two UCBs improves performance by mitigating the risk of overestimated CTRs from the LLM predictions.

After the fine-tuning of algorithms, we evaluate all algorithms on the test data set, which has never been used for either CTR training or algorithm tuning. Therefore, none of the algorithms know the true CTRs of the headlines in the test data in advance. We use the average CTR for each headline in the test data as the true CTRs to generate the click feedback in our numerical simulations. This approach of generating click outcomes is reasonable in our setting because each headline was tested on a large amount of traffic by Upworthy, resulting in relatively narrow confidence intervals.

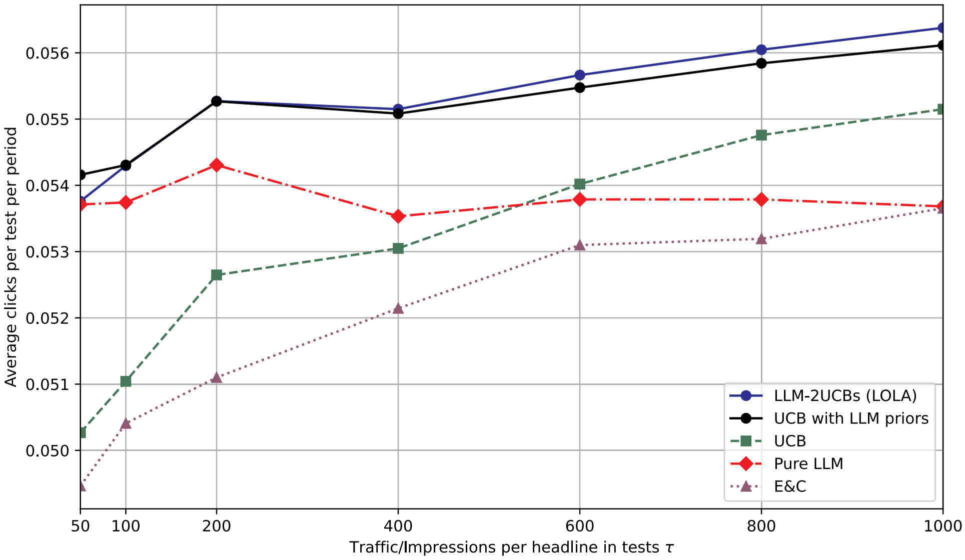
UCB and LOLA are both asymptotically sublinear in minimizing regret, which means there is no difference in average regret (R_T/T) between these two algorithms when the number of time horizons T goes to infinity, and the LLM-based prior information is not helpful in an infinite horizon case. However, in practice, the time horizon is never infinite. Therefore, we compare all the algorithms for different values of time horizons. One challenge in our setting is that the number of headlines varies across tests. Intuitively, for any given time horizon, it is always easier to minimize regret (or identify the best headline) when the A/B test has only two headlines compared with a case when there are three headlines, and of course, the problem only gets more challenging as the number of headlines increases. As a result, comparisons across time horizons are meaningful only when we keep the number of headlines constant. Because the number of headlines varies in our data set (see Table 3), we scale the time horizon for each A/B test using the metric "traffic/impressions per headline" (which we represent using the notion τ), where we choose τ from $\{50, 100, 200, 400, 600, 800, 1,000\}$. For example, if there are three headlines in an A/B test and the multiplier is $\tau = 100$, then we set $T = 100 \times 3 = 300$ for this test. Thus, the effective horizon across all tests for a given τ is the same. In our experiments, it will be particularly important to investigate the performance of different algorithms at small-medium multipliers because these represent situations where the media firm only has a small amount of traffic to work with and needs to adaptively experiment and effectively allocate impressions to the right headline with limited experimentation ability.

4.4. Numerical Results

We visualize the results from our numerical simulations in Figure 5[13] and document the pairwise comparison of different algorithms along with the relative percentage reward improvement and the p-values from the t-tests in Table 7. We find that, on average, LOLA performs the best among all algorithms across all time horizons.[14]

We now document a few additional patterns of interest. First, the pure-LLM approach exhibits consistent performance across different τ values because it is not an adaptive algorithm. The small fluctuation at $\tau \in \{50, 100, 200\}$ is because of the randomness of average clicks per period, which tends to be larger under the smaller time horizon. For the other three approaches, which actively learn the CTR during the experiments,

Figure 5. (Color online) Average Clicks per Experiment per Period Under Different Time Horizon Multipliers



Notes. Note that the y-axis captures the average clicks per test per period. For instance, if there is a test with two headlines receiving one and two clicks, respectively, under $\tau = 100$, then the average click per period in this test is calculated as $(1 + 2)/100 = 0.03$. The y value is simply the average of this number 0.03 over all tests. This measure scales well with the platform's total clicks in tests because headlines in different tests with different numbers of headlines take the same weight in this measure.

their average performance per period increases as the time horizon extends. This improvement is because the number of arms in the experiment remains constant, and with increasing periods, these algorithms have more opportunities to learn the CTRs, thus achieving better performance. Comparing the performance of LOLA with pure-LLM methods, we find that LOLA generally outperforms pure LLM except when $\tau = 50$, where there is no significant difference between LOLA and pure LLM. This is because the short planning horizon does not allow LOLA to learn new information over and above the information already available from the LLM. As τ increases to 1,000, we see that LOLA significantly outperforms pure LLM, generating 5.02% more clicks. This suggests that if the horizon is very short, the planner/firm will not see significant improvement over simply relying on LLM predictions. However, as the horizon increases, LOLA will start showing significant gains.

Second, comparing LOLA with regular UCB, we see that it always outperforms UCB. Interestingly, here, we see that as the time horizon increases, the relative improvement from LOLA diminishes. Intuitively, this is because both UCB and LOLA are active learning algorithms, and as the time horizon increases, they are both able to learn the CTRs better. Whereas LOLA starts with a relative information advantage, as the time horizon increases, this advantage diminishes. Formally, this pattern is because of the asymptotic sublinear of the UCB and LOLA algorithms, with the

Table 7. Percentage Improvement in Reward (Total Clicks) Between the Algorithms LOLA (LLM-2UCBs), UCB, Pure LLM, and E&C

Parameter τ	LOLA vs. UCB with LLM priors	LOLA vs. UCB	LOLA vs. Pure LLM	LOLA vs. E&C	UCB vs. Pure LLM	UCB vs. E&C	Pure LLM vs. E&C
50	−0.74**	6.95****	0.09	8.69****	−6.41****	1.62	8.59****
100	−0.02	6.38****	1.03**	7.72****	−5.02****	1.26	6.61****
200	0.01	4.98****	1.78****	8.16****	−3.05****	3.03**	6.27****
400	0.12	3.96****	3.02****	5.76****	−0.90	1.74*	2.66***
600	0.34	3.04****	3.49****	4.83****	0.43	1.73**	1.30
800	0.37*	2.35****	4.20****	5.36****	1.81***	2.94****	1.12
1,000	0.47**	2.23****	5.02****	5.08****	2.73****	2.79****	0.05

Notes. For the algorithm UCB with LLM priors, we only compare it to LOLA because they have similar performance, as shown in Figure 5. For example, under the impression per headline equal to 50, the average clicks per test of LOLA and UCB are 0.053760 and 0.050267, respectively, so the relative improvement is calculated as $0.053760/0.050267 - 1 = 6.95\%$. Parameter τ represents the average impression per headline to scale the time horizon $T = \tau \times K$.

$^{*}p \leq 0.05$; $^{**}p \leq 0.01$; $^{***}p \leq 0.001$; $^{****}p \leq 0.0001$.

same regret order $R_T/T = O(T^{-1/2})$ (see chapter 7.1 of Lattimore and Szepesvári 2020 for the proof), implying that as the time horizon approaches infinity, the initialization has a negligible effect on the average performance over an infinitely long period. We also see that E&C consistently underperforms compared with LOLA because it does not leverage LLM predictions and incurs larger regret with order $R_T/T = O(T^{-1/3})$; see chapter 6.1 of Lattimore and Szepesvári (2020) for the proof. Whereas E&C shows better performance as the horizon increases, even when τ = 1,000, LOLA is still better than E&C by 5.08%, which is a large improvement in the media and publishing industry.

Third, comparing the benchmark algorithms with each other, we see that when the horizon is short, the pure LLM does better, whereas UCB is better when the horizon is longer. In general, this reflects the intuition that active learning is not very useful when the time horizons are very short because the algorithm has no time to explore and learn the CTRs effectively (and then exploit this learning). So, simply going with the pure-LLM predictions is better. In contrast, when the horizons are longer, UCB (or active learning) outperforms using static predictions from the LLM. Note that LOLA combines the strengths of both UCB and pure LLM, and hence, it is able to outperform both and provide superior performance in both short and long horizons. We also note that E&C is generally worse than both pure LLM and UCB, which suggests that the current Upworthy practice is suboptimal. This is because E&C neither exploits LLM information, nor does it actively learn over time.

Finally, we will discuss the performance of UCB with LLM priors. Comparing LOLA (2-UCBs) with UCB with LLM priors, the key difference lies in whether the regular UCB U^1 is included in the algorithm. The core trade-off here is between the large potential error of U^2 when the LLM's predictions are incorrect and the stochastic noise in U^1, especially in the early stages. Our observations confirm this trade-off. In short time horizons $\tau = 50$, LOLA underperforms compared with UCB with LLM priors by a significant margin of −0.74%. This is because the error in the LLM-based CTR predictions is smaller than the stochastic noise of the regular CTR estimator at the start, and including the noisier U^1 reduces performance. However, as the time horizon increases (e.g., τ = 800 and τ = 1,000), LOLA significantly outperforms UCB with LLM priors by 0.37% and 0.47%, respectively. This improvement occurs because over longer periods, the error in LLM-based CTR predictions becomes more significant than the stochastic noise of the regular estimator, which can also be observed by comparing the performance between the standard UCB and pure-LLM methods. Incorporating the unbiased and less noisy U^1 helps enhance performance in these longer horizons. This observation suggests that it is especially beneficial to use 2-UCBs in longer horizons.

In summary, we find that LOLA outperforms existing approaches for experimentation by leveraging the strengths of LLMs and combining them with the benefits of active/online learning. In particular, our results demonstrate the value of LOLA as the time horizon increases, highlighting the limitations of relying solely on pure-LLM predictions or the naive E&C approach.

4.5. LOLA Variants and Extensions

So far, we have considered a version of LOLA that uses a fine-tuned CTR prediction model for LLM predictions in the first stage and an LLM-2UCBs algorithm that minimizes regret for online learning in the second step. However, LOLA is a general framework and can be easily adapted to a wide variety of settings. We highlight a few natural extensions and variants below. Details of the models and numerical results (when applicable) are shown in Web Appendix Section H.

4.5.1. Best Arm Identification. So far, we have focused on the goal of regret minimization, where the goal is to maximize the clicks/reward (per Equation (1)). Whereas regret minimization is the most commonly studied goal in active learning and the natural goal for businesses in most settings, another commonly studied goal is BAI. The goal of BAI problems is to identify the arm with the highest reward as fast and accurately as possible, regardless of accumulated regret. Formally, given a low failure rate δ, we aim to find the arm a^* with the highest expected reward, at least $1-\delta$ probability, while minimizing the number of total pulls.[15] We can easily modify the LOLA framework for a BAI goal. In Web Appendix Section H.1, we present an LLM-BAI algorithm that builds on the recently proposed BAI algorithm by Mason et al. (2020). This algorithm is particularly effective in our setting because it can accommodate scenarios where the difference between the rewards of arms is small or insignificant. We show that our LLM-BAI algorithm significantly outperforms both the plain BAI algorithm (proposed by Mason et al. 2020) as well as the standard A/B test under the same failure rate restriction. Please see Web Appendix Section H.1 for a detailed description of the LLM-BAI algorithm and its performance against the standard benchmarks.

4.5.2. Thompson Sampling. Our bandit specification in Section 4.1 reflects a stochastic bandit setting rather than a Bayesian bandit setting. However, our proposed algorithms can easily be applied in a Bayesian setting. We present a Thompson sampling version of our proposed LOLA algorithm (LLM-TS) and its numerical performance in Web Appendix Section H.2. We see that UCB-based algorithms perform better than

Thompson sampling–based algorithms in both the LLM-assisted version and the standard version. That is, LLM-TS performs worse than LLM-2UCBs, and standard TS performs worse than UCB. This pattern has been observed in the literature, and it is generally known that TS suffers from overexploration (Min et al. 2020); in other words, UCB's upper confidence shrinks faster, leading it to exploit more effectively. Nevertheless, in business settings where the firm already has a TS-based adaptive experimentation framework, it is easy to adapt the LOLA approach within this approach.

4.5.3. User Information. So far, we have not considered user-level features or how to personalize the content shown to different users based on their behavioral/context features (mostly because the Upworthy data do not have user-level features). Prior research has shown that using such features can significantly improve the match between users and content (Li et al. 2010, Yoganarasimhan 2020, Rafieian and Yoganarasimhan 2021). It is, however, easy to expand our LOLA approach to include user features in both the LLM training phase as well as the online learning phase. We present two versions of LOLA that extend existing contextual bandit algorithms to our setting. First, we can extend the standard contextual linear bandits (Chu et al. 2011) to our LOLA framework by incorporating both user and text features in the LLM training and online learning phases of LOLA. However, this linear bandit approach has limitations because it assumes that the rewards model is linear in text and user features. Therefore, we also propose another solution based on the FALCON algorithm that does not make any linearity assumption (Simchi-Levi and Xu 2022). We present the details of both these contextual LOLA algorithms and their pros and cons in Web Appendix Section H.3.

4.5.4. Alternative Rewards. In the Upworthy setting, the measure of reward is clicks. However, click-based metrics have raised concerns about promoting click-bait headlines, which may reduce user engagement or spread negativity. As a result, the news and media industry is now evolving to focus more on content quality and longer-term user engagement and retention metrics, although clicks still remain an important measure; see Reneau (2023) for a more detailed discussion. The LOLA framework can be easily adapted to alternative reward measures that align with new business goals, for example, time spent on the platform during a session or time spent on the article (and not just clicks on the headline).

Apart from these extensions, the LOLA approach is quite general and adaptable in other dimensions as well. For instance, it is agnostic to the exact nature of the LLM-based approach used in the first step. Any of the LLM-based approaches discussed in Section 3 can be used in the first step. This is important because the LLM models and fine-tuning approaches continue to advance quickly. Further, LOLA can also be used in conjunction with content/creatives created by LLMs themselves. For instance, one simple way to leverage fine-tuned models is to use them to generate content (instead of using human-generated headlines/content). Indeed, recent studies have shown that promotional content (ads/emails) generated by fine-tuned LLMs tend to outperform traditional personalized ads and human-generated content in effectiveness (Kumar and Kapoor 2023, Angelopoulos et al. 2024). LLM-generated headlines/content can be used as competing arms/treatments within our LOLA framework. Additionally, the LOLA framework can also be easily extended to content recommendation systems used by social media platforms such as X and Facebook. In summary, the plug-and-play nature of our approach can easily accommodate newer advances and developments at all stages of the implementation process.

5. Conclusion

In conclusion, in this paper, we examine if and how firms can leverage LLMs to enhance content experimentation in digital platforms. First, we examine how well LLMs can predict which content would be more appealing to users. We find that whereas LLMs provide informative predictions, even the best-performing fine-tuned LLMs are unable to perfectly predict which content will be more appealing to users and match the accuracy/regret of experimentation-based approaches. As such, firms cannot fully trust the prediction results solely from LLMs and replace experimentation.

We introduce LOLA, a novel experimentation framework that combines the predictive power of LLMs with the adaptive efficiency of online learning algorithms to enhance content experimentation. By leveraging LLM-based prediction models as priors, our approach minimizes regret in real time, leading to significant improvements in total reward. Comprehensive numerical experiments based on a large-scale A/B testing data set demonstrate that LOLA outperforms the traditional A/B tests and pure online learning algorithms.

From a managerial perspective, LOLA offers a versatile solution to a broad set of scenarios where the firm needs to decide which content to display to users. LOLA is particularly valuable under limited traffic conditions that are common in many digital media platforms. This includes cases where content becomes stale quickly (e.g., news articles) or social media platforms such as X and TikTok, where users generate a lot of new content regularly, and as a result, the amount of content is high relative to the amount of traffic. Whereas our focal application is in the context of headline selection in the news and media industry, the framework can easily be applied to other problems such as digital advertising, email marketing, and promotions. Indeed, a large literature in marketing focuses on how to use experiments in conjunction with machine learning methods to identify optimal treatments and/or personalize treatments; see Rafieian and Yoganarasimhan (2023) for a detailed review. The LOLA approach can be easily used in many of those settings and help with maximizing outcomes of interest while simultaneously reducing the cost of experimentation.

From an implementation perspective, LOLA offers many advantages. Because it can utilize open-source LLM models, it is cost-effective and can be easily deployed across different tasks once LLM models are fine-tuned in relevant data sets. The fine-tuning process itself is cost-efficient and can be performed on entry-level GPUs, making advanced LLM-based techniques accessible to firms with limited budgets. Relying on open-source LLMs also ensures that the firm's data stay within the firm and are not accessible to the owners of proprietary LLM models (such as Google and OpenAI). Further, our approach is compatible with many bandit-based experimental systems used by large digital firms such as Amazon (Fiez et al. 2024) and, as such, is easy to integrate and deploy.

Our study paves the way for future research into developing more sophisticated and integrated frameworks that effectively balance the predictive strengths of LLMs with the adaptability of bandit algorithms. For instance, our current 2-UCB strategy employs a uniform n^{aux} across all arms, which assumes an equal level of uncertainty in CTR predictions used as priors. However, an alternative approach could involve calibrating different uncertainty levels for different arms, potentially leading to improved performance. More generally, we expect the intersection of LLMs and bandits to be a promising avenue of future research and hope that the ideas presented in this study can be expanded and developed in different directions.

Acknowledgments

The authors thank Yuyan Wang and Yuting Zhu for detailed feedback that has improved the paper. The authors also thank the participants of the China India Insights Conference 2024, HKU Marketing Camp 2024, Informs Marketing Science 2024, QME 2024, USC Marketing seminar, and UW–UBC Conference 2024 for feedback. The code used in this study is publicly available at LLM News Github Repository (https://github.com/DDDOH/LLM_News) to enable the reproduction of the authors' results and to support future research that applies LLMs to experimentation and business problems. The author(s) have no competing interests to declare.

Endnotes

[1] We use the terms adaptive experimentation, online learning algorithm, and bandits interchangeably throughout the paper.

[2] Similarly, in the context of digital advertising, Schwartz et al. (2017) use field experiments to demonstrate that adaptive algorithms outperform the standard balanced A/B test.

[3] Note that just because an average reader cannot discern/identify the most engaging or catchiest headline among a pair/set of headlines, it does not follow that such differences do not exist. Indeed, the whole premise of the experimentation culture in digital firms relies on the idea that whereas managers/lay users cannot correctly predict which version of an ad creative/promotion/website layout will be more engaging, such differences exist and that identifying these differences and implementing the best-performing arm is important and will lead to better consumer engagement and business outcomes.

[4] Note that this test set is slightly smaller than 20% of headline tests. This is because, to avoid information leakage, we deleted the duplicated tests that had the same headlines in the training data set. This happens because Upworthy allowed additional experiments for the same headline under a new experiment ID, and the editor may require an extra test to make an informed decision.

[5] The temperature parameter scales and controls the randomness of the model's responses. A larger value results in a more diverse and creative output, whereas a lower temperature makes the output more deterministic and focused. The default setting with temperature is 1.0. We also tested the default temperature, and we did not observe any significant difference in accuracy.

[6] As of May 24, 2024, OpenAI recommends defaulting to GPT-3.5-turbo, GPT-4-turbo, or GPT-4o. OpenAI notes that GPT-4-turbo and GPT-4o offer similar levels of intelligence.

[7] We did run experiments with a larger number of demonstrations but did not find any significant improvements from doing so and hence do not report them here.

[8] Embedding models and standard prompt-based LLMs share some common features, for example, the underlying transformer architectures and extensive pretraining on large text corpora. However, they are optimized for different tasks. Models such as GPT are optimized for generating coherent and contextually relevant text based on input prompts, whereas embedding models focus on producing high-quality embeddings for downstream tasks.

[9] Some proprietary LLMs offer black-box fine-tuning. For instance, OpenAI allows fine-tuning GPT models through its API in a black-box manner. Users provide the training and test data sets and select the base GPT model but are not provided with any details of the fine-tuning process or the underlying model parameters. This approach also has a number of other drawbacks: the fine-tuning API is quite expensive compared to the open-source fine-tuning approaches, the fine-tuned model cannot be hosted on a local machine, and there are data privacy concerns because the media firm's proprietary data will need to be submitted to OpenAI. Given these drawbacks, it is preferable for firms to fine-tune open-source LLMs, as the process and data are fully under the control of the firm. However, for the completeness of comparison, we fine-tuned the GPT-4o (`gpt-4o-2024-08-06` version) model on the training set using this API, with the default configuration provided by OpenAI and the same training and test data sets. Input to the model is several headlines of one article, and the target output is the index of the catchiest headline. We obtained an accuracy of 48.82% on the test set, but this result comes with a high cost of $71.77 for the test set alone.

[10] There are many alternatives for PEFT, such as Prefix-Tuning (Li and Liang 2021), Adapter Layers (Houlsby et al. 2019), and BitFit (Zaken et al. 2021). We select LoRA because of its excellent balance between simplicity and efficiency, making it the top choice for fine-tuning.

[11] This concern is somewhat assuaged by Web Appendix Section C.4, which shows that consumer preferences did not change significantly

during the 2013–2015 time frame (at least within the prediction capabilities of the LLMs). However, we cannot empirically comment on the period beyond 2015.

[12] Note that because this is a pure prediction-based greedy approach, there are no hyperparameters associated with the algorithm, and hence, we do not use the fine-tuning data.

[13] The y metric in Figure 5 is essentially the accumulated reward divided by τ, that is, the value of the x-axis. We choose to report this y metric instead of the actual regret for two reasons: (1) the ground truth arm is not known in practice, making it impossible to compute regret, and (2) for better visualization. Specifically, the accumulated reward (unscaled by τ) grows very rapidly as τ increases, which compresses the visual differences between algorithms, making them difficult to distinguish with the naked eye. With this being said, the results in Table 7 can still be interpreted as the percentage improvement of the accumulated reward because all algorithms are measured at the fixed τ and the accumulated reward equals $Y \times \tau$.

[14] Note that we cannot translate the click improvement to revenues or other business outcomes of interest because we do not have data on how clicks/time spent on the website relate to downstream metrics such as advertising and subscription revenues. Therefore, we do not make additional claims on the business impact of these numbers. That said, if the firm has access to these numbers, it should be relatively straightforward to make that connection.

[15] This is also called fixed-confidence setting (Garivier and Kaufmann 2016). Another common approach is the fixed-budget setting, where the goal is to maximize the probability of finding the best arm within a fixed number of pulls.

References

Angelopoulos P, Lee K, Misra S (2024) Causal alignment: Augmenting language models with A/B tests. Preprint, submitted April 15, http://dx.doi.org/10.2139/ssrn.4781850.

Aramayo N, Schiappacasse M, Goic M (2023) A multiarmed bandit approach for house ads recommendations. *Marketing Sci.* 42(2): 271–292.

Auer P, Cesa-Bianchi N, Fischer P (2002) Finite-time analysis of the multiarmed bandit problem. *Machine Learning* 47:235–256.

Banerjee A, Urminsky O (2024) The language that drives engagement: A systematic large-scale analysis of headline experiments. *Marketing Sci.*, ePub ahead of print November 4, https://doi.org/10.1287/mksc.2021.0018.

Brand J, Israeli A, Ngwe D (2023) Using LLMs for market research. Preprint, submitted March 30, http://dx.doi.org/10.2139/ssrn.4395751.

Brucks M, Toubia O (2023) Prompt architecture can induce methodological artifacts in large language models. Preprint, submitted June 25, http://dx.doi.org/10.2139/ssrn.4484416.

Bubeck S, Liu C-Y (2013) Prior-free and prior-dependent regret bounds for Thompson sampling. *NIPS'13: Proc. 27th Internat. Conf. Neural Inform. Processing Systems*, vol. 1 (Curran Associates Inc., Red Hook, NY), 638–646.

Chu W, Li L, Reyzin L, Schapire R (2011) Contextual bandits with linear payoff functions. Geoffrey G, David D, Miroslav D, eds. *Proc. 14th Internat. Conf. Artificial Intelligence Statist.*, vol. 15 (PMLR, New York), 208–214.

Coenen A (2019) How New York Times is experimenting with recommendation algorithms. *NYT Open* (October 17), https://open.nytimes.com/how-the-new-york-times-is-experimenting-with-recommendation-algorithms-562f78624d26.

Fiez T, Nassif H, Chen Y-C, Gamez S, Jain L (2024) Best of three worlds: Adaptive experimentation for digital marketing in practice. *Proc. ACM Web Conf. 2024* (Association for Computing Machinery, New York), 3586–3597

Garivier A, Kaufmann E (2016) Optimal best arm identification with fixed confidence. Vitaly F, Alexander R, Ohad S, eds. *Conf. Learning Theory*, vol 49 (PMLR, New York), 998–1027.

Gui G, Toubia O (2023) The challenge of using LLMs to simulate human behavior: A causal inference perspective. Preprint, submitted December 24, https://arxiv.org/abs/2312.15524.

Gur Y, Momeni A (2022) Adaptive sequential experiments with unknown information arrival processes. *Manufacturing Service Oper. Management* 24(5):2666–2684.

Hauser JR, Urban GL, Liberali G, Braun M (2009) Website morphing. *Marketing Sci.* 28(2):202–223.

Houlsby N, Giurgiu A, Jastrzebski S, Morrone B, De Laroussilhe Q, Gesmundo A, Attariyan M, Gelly S (2019) Parameter-efficient transfer learning for NLP. Kamalika C, Ruslan S, eds. *Proc. 36th Internat. Conf. Machine Learning*, vol. 97 (PMLR, New York), 2790–2799.

Hu EJ, Shen Y, Wallis P, Allen-Zhu Z, Li Y, Wang S, Wang L, Chen W (2022) LoRA: Low-rank adaptation of large language models. *Conf. Paper at ICLR 2022*, vol. 2 (OpenReview.net), 3.

Jain L, Li Z, Loghmani E, Mason B, Yoganarasimhan H (2024) Effective adaptive exploration of prices and promotions in choice-based demand models. *Marketing Sci.* 43(5):1002–1030.

Kumar M, Kapoor A (2023) Generative AI and personalized video advertisements. Preprint, submitted November 17, http://dx.doi.org/10.2139/ssrn.4614118.

Lattimore T, Szepesvári C (2020) *Bandit Algorithms* (Cambridge University Press, Cambridge, UK).

Li XL, Liang P (2021) Prefix-tuning: Optimizing continuous prompts for generation. Preprint, submitted January 1, https://arxiv.org/abs/2101.00190.

Li L, Chu W, Langford J, Schapire RE (2010) A contextual-bandit approach to personalized news article recommendation. *WWW'10: Proc. 19th Internat. Conf. World Wide Web* (Association for Computing Machinery, New York), 661–670.

Li P, Castelo N, Katona Z, Sarvary M (2024) Frontiers: Determining the validity of large language models for automated perceptual analysis. *Marketing Sci.* 43(2):254–266.

Liberali G, Ferecatu A (2022) Morphing for consumer dynamics: Bandits meet hidden Markov models. *Marketing Sci.* 41(4): 769–794.

Mason B, Jain L, Tripathy A, Nowak R (2020) Finding all ϵ-good arms in stochastic bandits. *NIPS'20: Proc. 34th Internat. Conf. Neural Inform. Processing Systems* (Curran Associates Inc., Red Hook, NY), 20707–20718.

Matias JN, Munger K, Le Quere MA, Ebersole C (2021) The Upworthy Research Archive, a time series of 32,487 experiments in US media. *Sci. Data* 8(1):195.

Min S, Moallemi CC, Russo DJ (2020) Policy gradient optimization of Thompson sampling policies. Preprint, submitted June 30, https://arxiv.org/abs/2006.16507.

Min S, Lyu X, Holtzman A, Artetxe M, Lewis M, Hajishirzi H, Zettlemoyer L (2022) Rethinking the role of demonstrations: What makes in-context learning work? Preprint, submitted February 25, https://arxiv.org/abs/2202.12837.

Misra K, Schwartz EM, Abernethy J (2019) Dynamic online pricing with incomplete information using multiarmed bandit experiments. *Marketing Sci.* 38(2):226–252.

Patil R, Boit S, Gudivada V, Nandigam J (2023) A survey of text representation and embedding techniques in NLP. *IEEE Access* 11:36120–36146.

Rafieian O, Yoganarasimhan H (2021) Targeting and privacy in mobile advertising. *Marketing Sci.* 40(2):193–218.

Rafieian O, Yoganarasimhan H (2023) AI and personalization. *Artificial Intelligence in Marketing* (Emerald Publishing Limited, Leeds, UK), 77–102.

Reneau A (2023) Study of Upworthy headlines claims negativity drives website clicks. We have some thoughts. *Upworthy*

(March 22), https://www.upworthy.com/upworthy-negative-headlines-study.

Russo D, Van Roy B (2014) Learning to optimize via posterior sampling. *Math. Oper. Res.* 39(4):1221–1243.

Saunshi N, Malladi S, Arora S (2020) A mathematical exploration of why language models help solve downstream tasks. Preprint, submitted October 7, https://arxiv.org/abs/2010.03648.

Schulhoff S, Ilie M, Balepur N, Kahadze K, Liu A, Si C, Li Y, et al. (2024) The prompt report: A systematic survey of prompting techniques. Preprint, submitted June 6, https://arxiv.org/abs/2406.06608.

Schwartz EM, Bradlow ET, Fader PS (2017) Customer acquisition via display advertising using multi-armed bandit experiments. *Marketing Sci.* 36(4):500–522.

Simchi-Levi D, Xu Y (2022) Bypassing the monster: A faster and simpler optimal algorithm for contextual bandits under realizability. *Math. Oper. Res.* 47(3):1904–1931.

Symonds A (2017) When a headline makes headlines of its own. *New York Times* (March 23), https://www.nytimes.com/2017/03/23/insider/headline-trump-time-interview.html.

Xie SM, Min S (2022) How does in-context learning work? A framework for understanding the differences from traditional supervised learning. *SAIL Blog* (August 1), http://ai.stanford.edu/blog/understanding-incontext/.

Xie SM, Raghunathan A, Liang P, Ma T (2021) An explanation of in-context learning as implicit Bayesian inference. Preprint, submitted November 3, https://arxiv.org/abs/2111.02080.

Yang K (2024) Milestones on our journey to standardize experimentation at New York Times. *NYT Open* (March 26), https://open.nytimes.com/milestones-on-our-journey-to-standardize-experimentation-at-the-new-york-times-2c6d32db0281.

Yoganarasimhan H (2020) Search personalization using machine learning. *Management Sci.* 66(3):1045–1070.

Yoganarasimhan H, Yakovetskaya I (2024) From feeds to inboxes: A comparative study of polarization in Facebook and email news sharing. *Management Sci.* 70(9):6461–6472.

Zaken EB, Ravfogel S, Goldberg Y (2021) Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. Preprint, submitted June 18, https://arxiv.org/abs/2106.10199.

Zhao J, Wang T, Abid W, Angus G, Garg A, Kinnison J, Sherstinsky A, Molino P, Addair T, Rishi D (2024) LoRA land: 310 fine-tuned LLMs that rival GPT-4, a technical report. Preprint, submitted April 29, https://arxiv.org/abs/2405.00732.