LOLA: LLM-Assisted Online Learning Algorithm for Content Experiments

Hema Yoganarasimhan

University of Washington

Joint work with

Zikun Ye (UW) and Yufeng Zheng (U Toronto)

Content Experiments

The New York Times

yahoo!news

- **Media Firms Need**: Automated and efficient experimentation methods to evaluate the quality of their content in the digital setting.
- **Content Type**: Article, headline, etc.
- **Goal**: Find best content that maximizes user engagement (clicks).
- **Current Methods**: A/B test, online learning algorithm (bandit).

Headline Experiment Example

9 Out Of 10 Americans Are Completely Wrong About This Mind-Blowing Fact

MUST WATCH: Do You Want To Take The Red Pill Or The Blue Pill? Don't Say I Didn't Warn You.

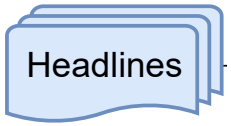
Meet The Person Who Has No Clue How Money Really Works. Ok, Stop Looking In The Mirror Now.

Upworthy's objective: Identify the best headline without wasting too much traffic on poorly performing headlines

Two Standard Solutions

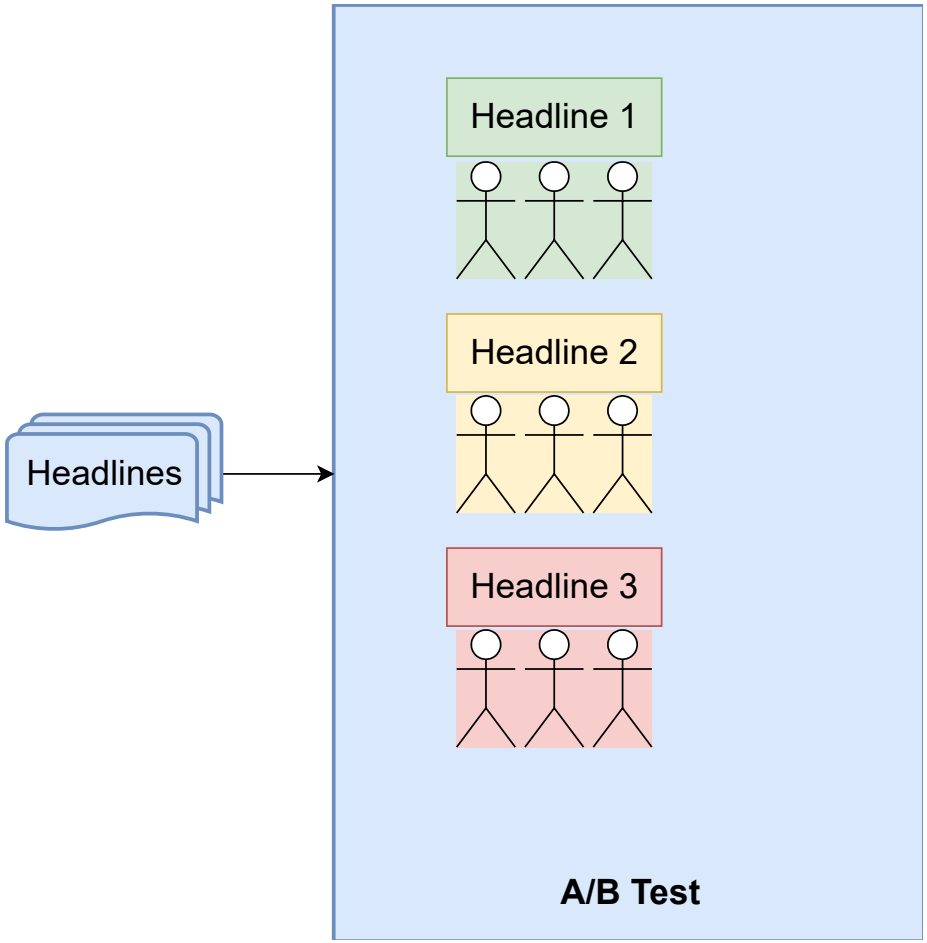
- A/B tests
  - Also referred to as the Explore and Commit Strategy (E&C)
- Bandits or Online Algorithms

Solution I - Standard A/B Test



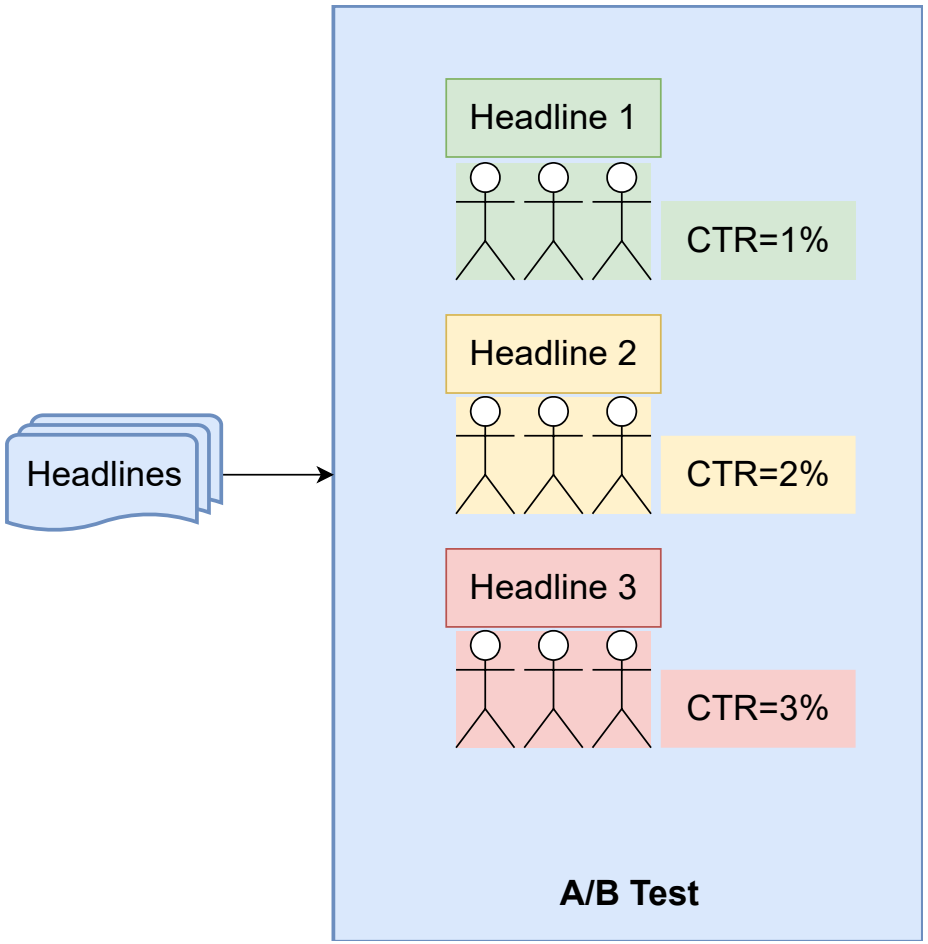
Step1: Editors come up with multiple headlines for the same article.

Solution I - Standard A/B Test



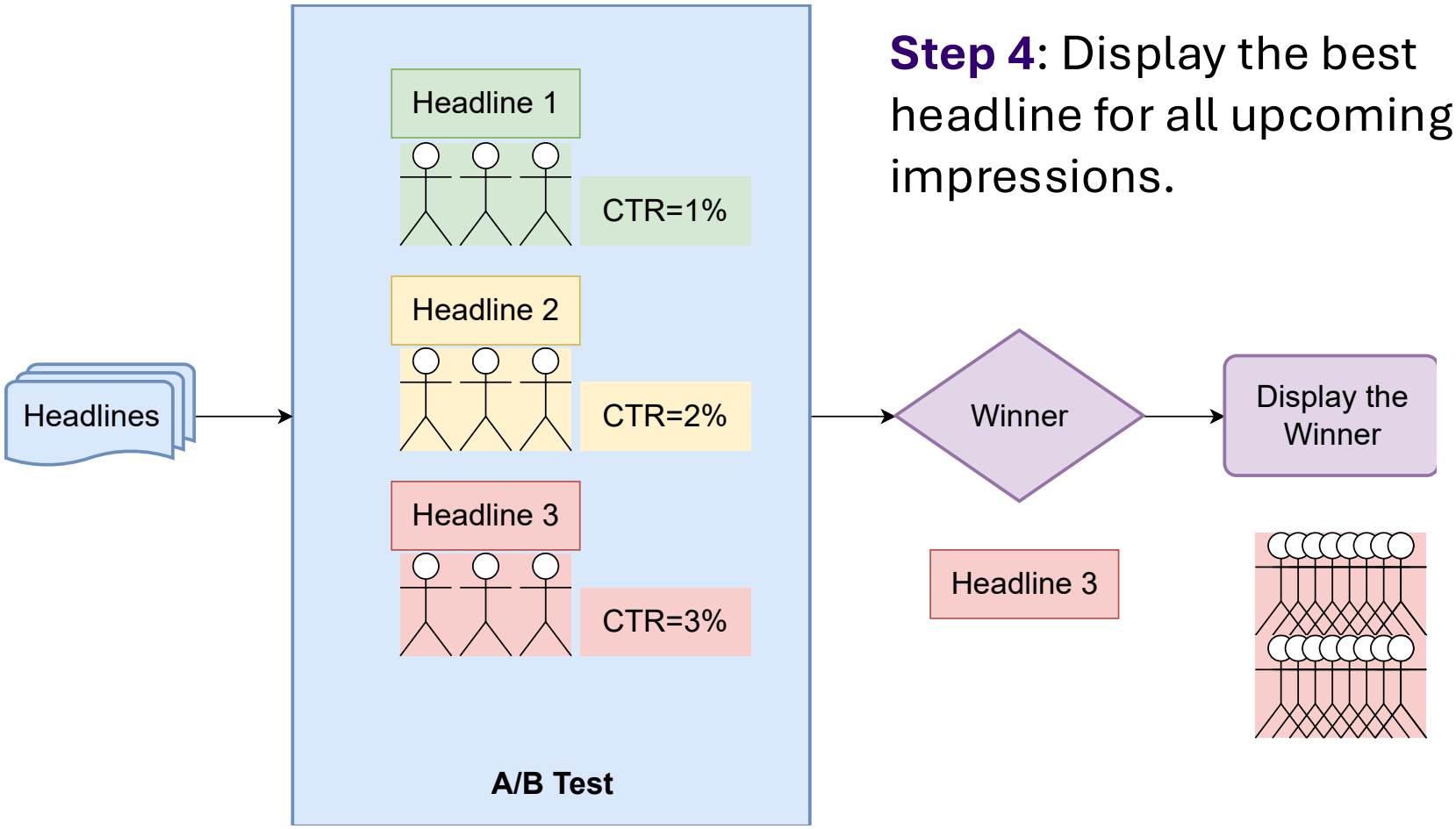
Step 2: Randomly assign users to see one of these headlines with the equal probability.

Solution I - Standard A/B Test



Step 3: Collect data and estimate CTR.

Solution I - Standard A/B Test



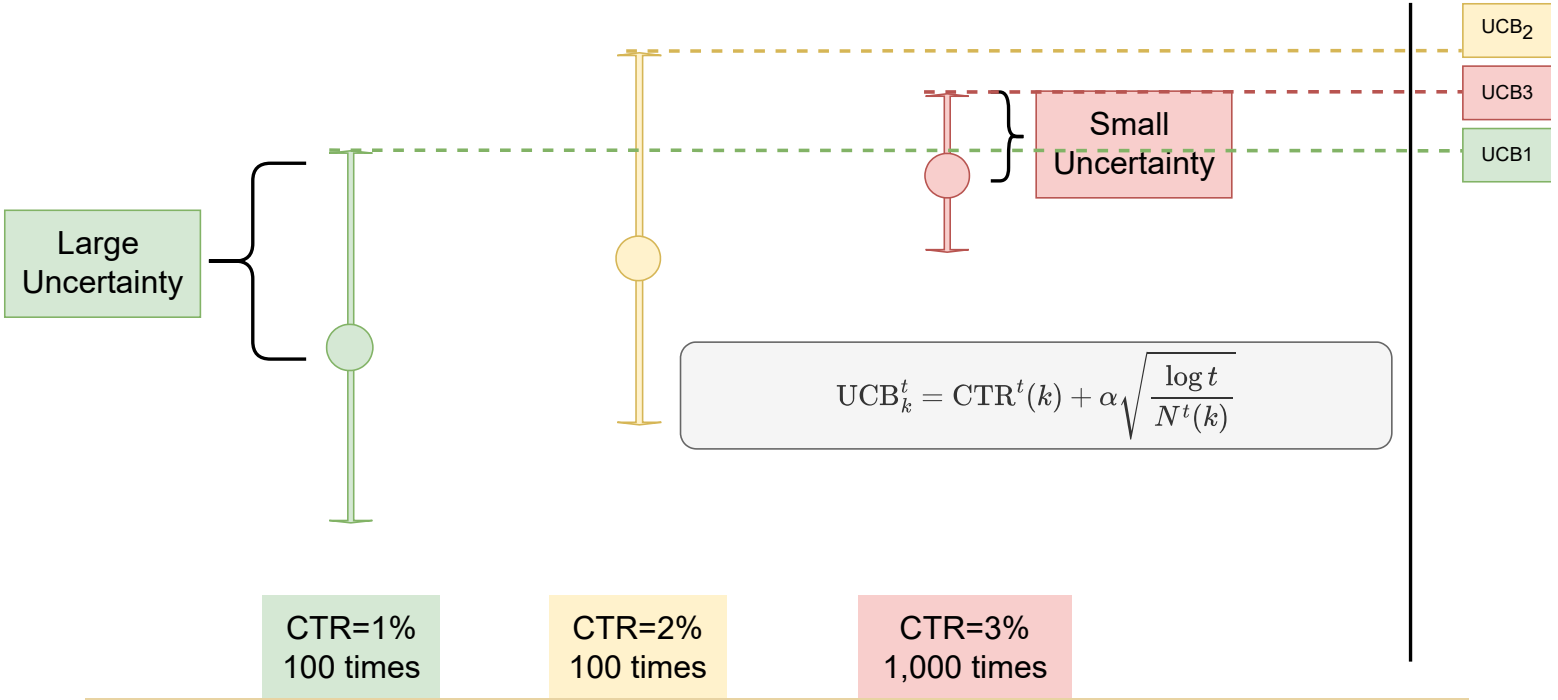
Step 4: Display the best headline for all upcoming impressions.

A/B Test - Pros and Cons

- **Pros:** Gold standard, trustworthy, and accurate
- **Cons:** Waste of impressions on bad headlines
  - Problematic in the news industry since news articles tend to have short lifetimes and become stale within a day or two.
  - 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.
- Nevertheless, used in practice by many firms, e.g., Upworthy

Solution II - Online Learning Algorithm

- Also known as adaptive experiments or bandits
- **Intuition:** Smartly balance exploration and exploitation



UCB policy: at each time t, play the arm k with the highest UCB_k^t

The New York Times yahoo!news

Used by NYTimes and Yahoo!

L. Li, W. Chu, J. Langford, and R. E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 661–670, 2010.

Online Learning - Pros and Cons

- **Pros:** Less traffic wasted on bad headlines compared to standard A/B tests.
  - Reduce the extent of exploration over time by moving traffic away from poorly performing treatments dynamically.
  - Typically outperform A/B tests on regret (can be shown theoretically as well as empirically)

- **Cons:** Cold start problem
  - Online learning typically assumes all items are equally good initially.
  - It is still costly to run the online learning algorithm.
  - Can inadvertently converge on sub-optimal arm, especially when traffic is small.

Solution III – Large Language Models?

Question: Can we bypass the experimentation and simply let the LLM decide which content is better?

Recent research has shown that LLM is good at understanding news content (Yoganarasimhan and Yakovetskaya, 2023).

LLM can mimic human preferences: Automated perceptual analysis (Li et al., 2024), Consumers' WTP (Brand et al., 2023)

Pros of LLM: no experiment needed, as experiment can be costly in news industry.

We leverage a large-scale dataset from Upworthy to answer this question.

Outline

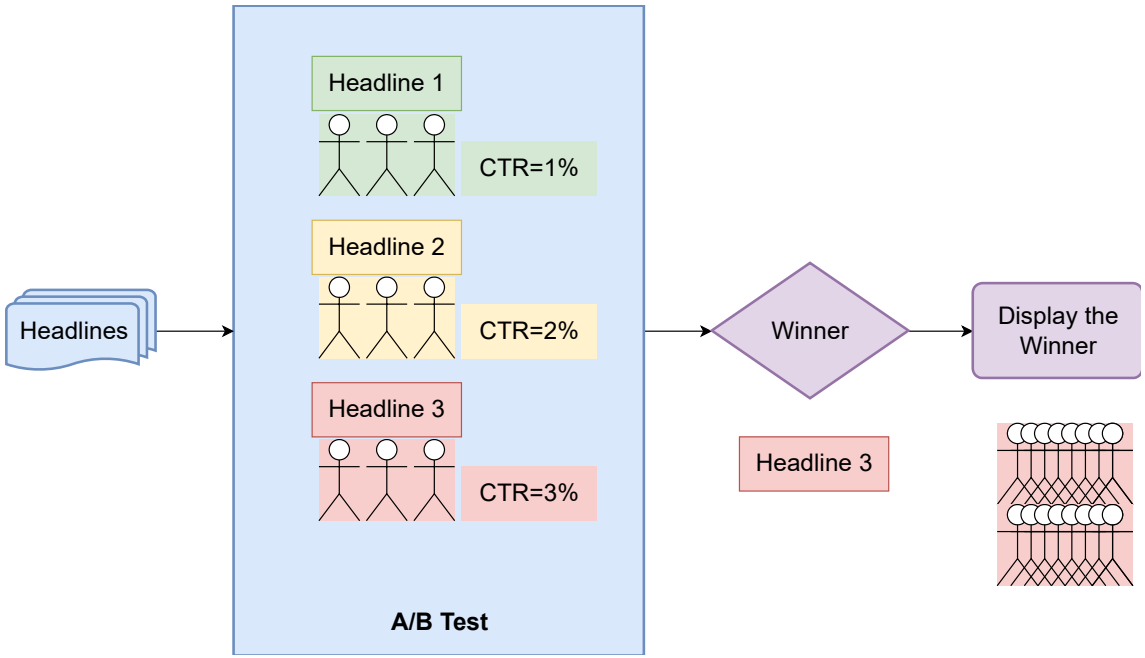
Upworthy Data

Pure LLM-based Approaches: Prompt, Embedding, and Fine-Tuning

Our Approach: LOLA Framework and Performance

Upworthy Data

- Upworthy is a media firm, founded in 2012.
- Upworthy recorded and released its A/B test data from 2013 to 2015.
- ~23% of impressions (2.4 billion site-wide) were used for A/B tests.
- ~54% of A/B tests were on headlines.



The data is public available at https://osf.io/jd64p/

Upworthy Data - Headline Tests

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

18k
Headline A/B Tests

77k
Headlines

4.4
Avg Headlines/Test

277 Million
Impressions

3.7 Million
Clicks

Outline

Upworthy Data

Pure LLM-based Approaches: Prompt, Embedding, and Fine-Tuning

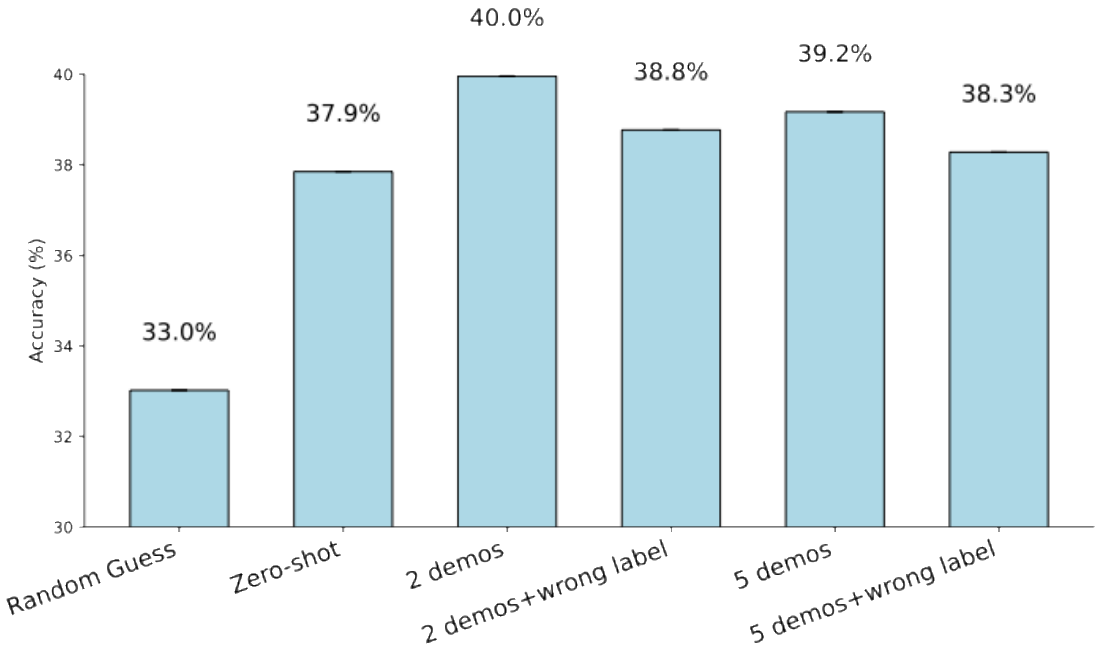
Our Approach: LOLA Framework and Performance

LLM-based Approaches

- **Goal** – Use LLMs to identify the best headline and assign all the traffic to it
  - No Experimentation

- We consider three widely used LLM-based methods
  - Prompt based approaches
  - Embedding-based approaches
  - Fine-tuning

Results for Prompt Based Approaches



- **Poor** performance with all accuracies **< 40%**
- In-context learning (2 demos) outperforms zero-shot prompting
- More demos do **NOT** help
- Input-output mapping is **NOT** important in demos

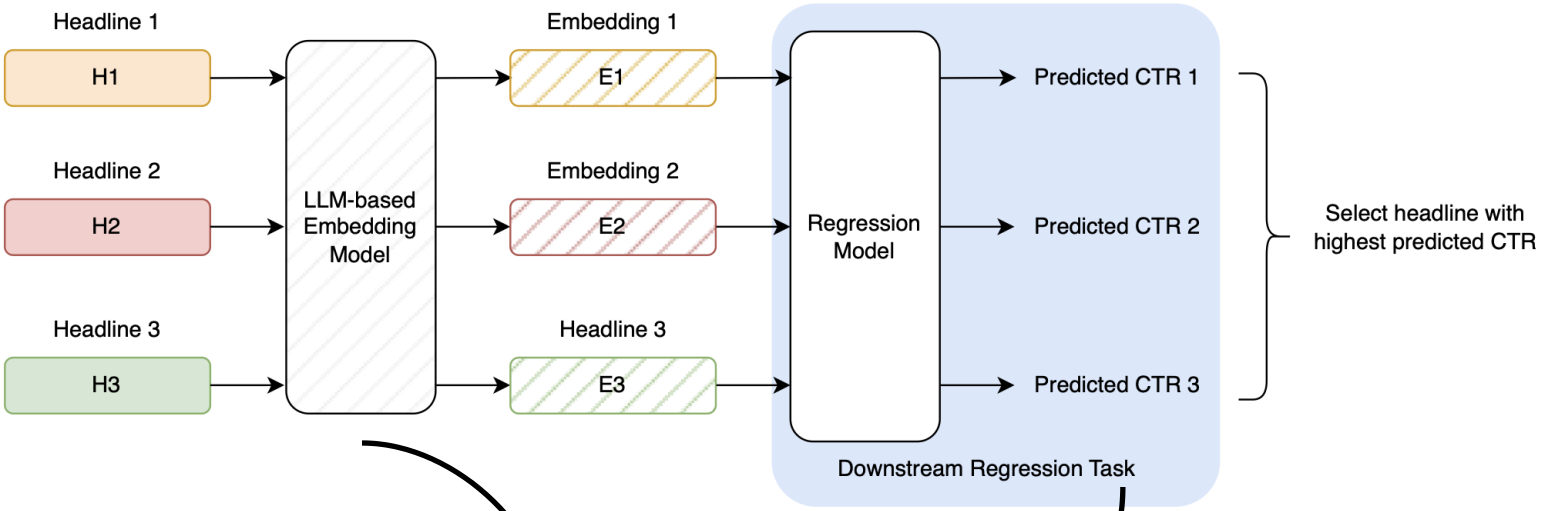
Text Embeddings

- **Text embeddings:** a numerical representation that captures the contextual information of the text (Patil et al., 2023)

- Embedding model transforms words, sentences, or entire documents into vectors of real numbers.

- Useful for many downstream tasks:
  - Classification
  - Prediction (CTR)
  - Clustering
  - Recommendation
  - …

Embedding + Predictive Model



OpenAI's best embedding model: **text-embedding-3-large**

This model is **NOT** trainable.

Linear Regression

Multilayer Perceptron

Results of the Embedding-Based Method

Embedding Model	CTR Prediction Model	Accuracy on all test data
OpenAI-256E	Linear	**46.28%**
	MLP	44.38%
OpenAI-3072E	Linear	43.06%
	MLP	45.17%

- 12,376 headline tests as training set and 3,263 headline tests as test set
- Significantly outperforms prompt-based approaches (40% and 66%)
- MLP has similar performance as the simple linear regression
- Larger embeddings do not improve performance
- Robust and transparent: We do not need to go back and forth in GPT conversations. Can use open-sourced LLMs to get embeddings.

Note: We also consider other embedding models: (1) word2vec, (2) Llama-3.

Fine-Tuning

- So far, we did not touch the LLMs' parameters.
  - Prompt-based approach: **ask** LLMs
  - Embedding-based approach: **use** LLMs to get embeddings

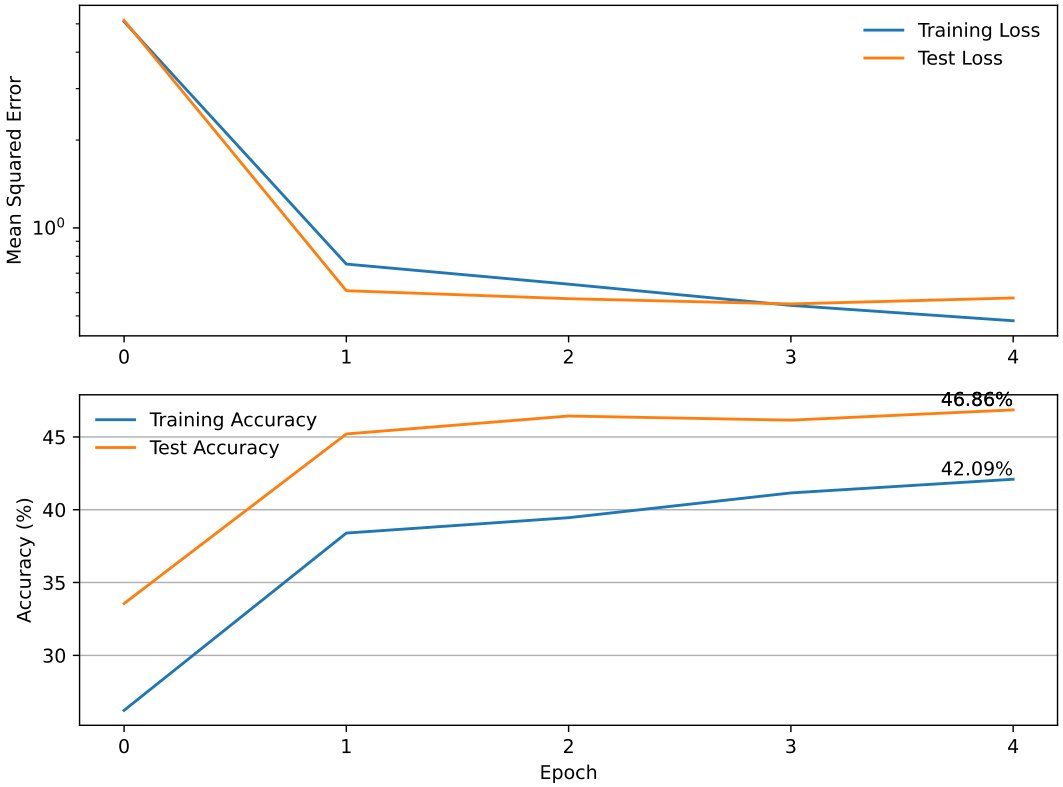
- Maybe updating the parameters of LLMs using our data can improve our performance.

- Cannot update all parameters $\Longrightarrow$ Update only a small portion (Parameter Efficient Fine-Tuning)
  - Low-rank fine-tuning (Hu et al, 2021)

- Researchers have done this for many use cases: https://predibase.com/fine-tuning-index

E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Performance of LoRA



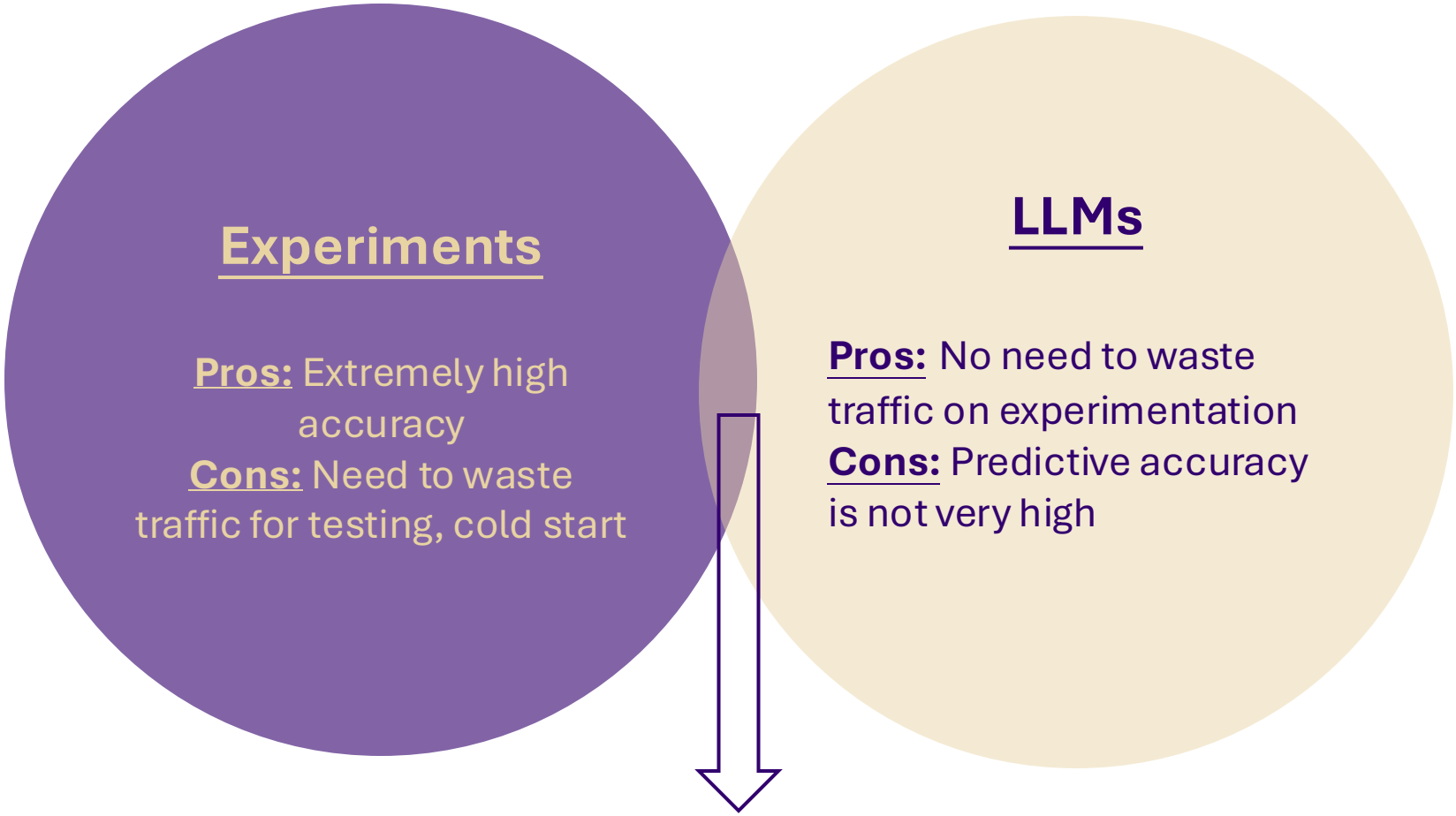
- 46% accuracy on test data (only marginally better than embedding-based approaches)

So far

Pure LLM-based methods are informative but not accurate.

Can we do better? Can we combine LLMs with experiments?

Our Approach



LOLA: LLM-Assisted Online Learning Algorithm

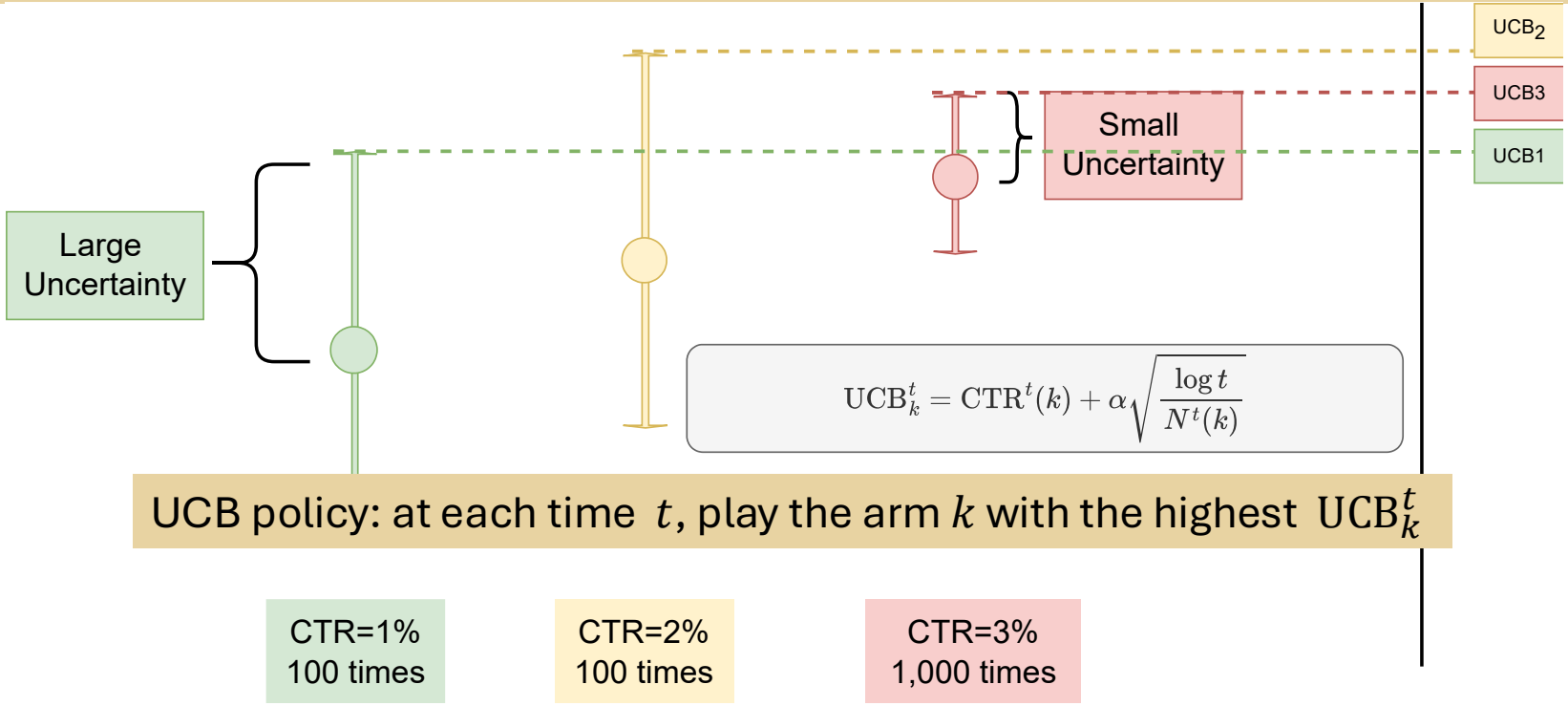
Key idea: Combine the benefits of LLMs with the strengths of experimentation

LOLA

- **Key Idea**
  - Use predictions from LLMs (of CTRs) as priors in the online experiment
- **Key Challenges**
  - No theoretical/empirical measures of the “errors” in LLM predictions.
  - How should we weight the LLM predictions vs. data from my live bandit/experiment?
- **Two step procedure**
  - **Step 1:** Use any LLM-based method to obtain an estimate of the CTR of each headline/arm
  - **Step 2:** Warm start the online experiment/bandit using the LLM-based priors

Upper Confidence Bound (UCB)

Idea of UCB: adaptively learn the CTR and make the decision in a smart way.



So far at time $t = 1201$

- Arm 1 has been played 100 times with 1 click, $\mathrm{CTR}^t(1) = 1\%$, $N^t(1) = 100$
- Arm 2 has been played 100 times with 2 clicks, $\mathrm{CTR}^t(2) = 2\%$, $N^t(2) = 100$
- Arm 3 has been played 1000 times with 30 clicks, $\mathrm{CTR}^t(3) = 3\%$, $N^t(3) = 1000$

Our Approach LOLA: LLM + UCB

However, at the very beginning $t = 1$, we have **no prior information** about CTRs, and have to initialize the CTRs as equally good, which wastes traffic.

At time $t = 1$?

- Arm 1 has been played 0 times with 0 click, $\mathrm{CTR}^t(1) = ?\,\%$, $N^t(1) = 0$
- Arm 2 has been played 0 times with 0 click, $\mathrm{CTR}^t(2) = ?\,\%$, $N^t(2) = 0$
- Arm 3 has been played 0 times with 0 click, $\mathrm{CTR}^t(3) = ?\,\%$, $N^t(3) = 0$

Can LLM help? Because we know LLM is informative of which headline is better.

- We use LLM to predict CTRs at the very beginning.
- Arm 1 has been played 0 times with 0 click, $\mathrm{CTR}^t(1) = \mathbf{0.9}\%$, $N^t(1) = 0$
- Arm 2 has been played 0 times with 0 click, $\mathrm{CTR}^t(2) = \mathbf{2.1}\%$, $N^t(2) = 0$
- Arm 3 has been played 0 times with 0 click, $\mathrm{CTR}^t(3) = \mathbf{2.7}\%$, $N^t(3) = 0$

Our Approach LOLA: LLM + UCB

At time $t = 1$, we use LLM to predict CTRs.

- Arm 1 has been played 0 times with 0 click, $\mathrm{CTR}^t(1) = \mathbf{0.9\%}$, $N^t(1) = 0$

But how to update CTRs later on?

- For example, at certain time, Arm 1 has been played 10 times with 1 click.
- Then the regular CTR estimator in UCB $= \mathbf{10\%}$. How to leverage LLM's prediction of $\mathbf{0.9\%}$?

We come up with **"LLM equivalent size".** If LLM prediction's equivalent size is **1000**, then $\mathbf{0.9\%}$ translates to "1000 **extra** times with 9 **extra** clicks" played by **LLM**

- **LOLA:** Arm 1 has been played 1010 times with 10 clicks, $\mathrm{CTR}^t(1) = \mathbf{0.99\%}$, $N^t(1) = 1010$
- **UCB:** Arm 1 has been played 10 times with 1 click, $\mathrm{CTR}^t(1) = \mathbf{10\%}$, $N^t(1) = 10$
- **Pure LLM:** $\mathrm{CTR}^t(1) = \mathbf{0.9\%}$

We basically **warm start** the UCB algorithm by using LLM CTR predictions as **priors**.

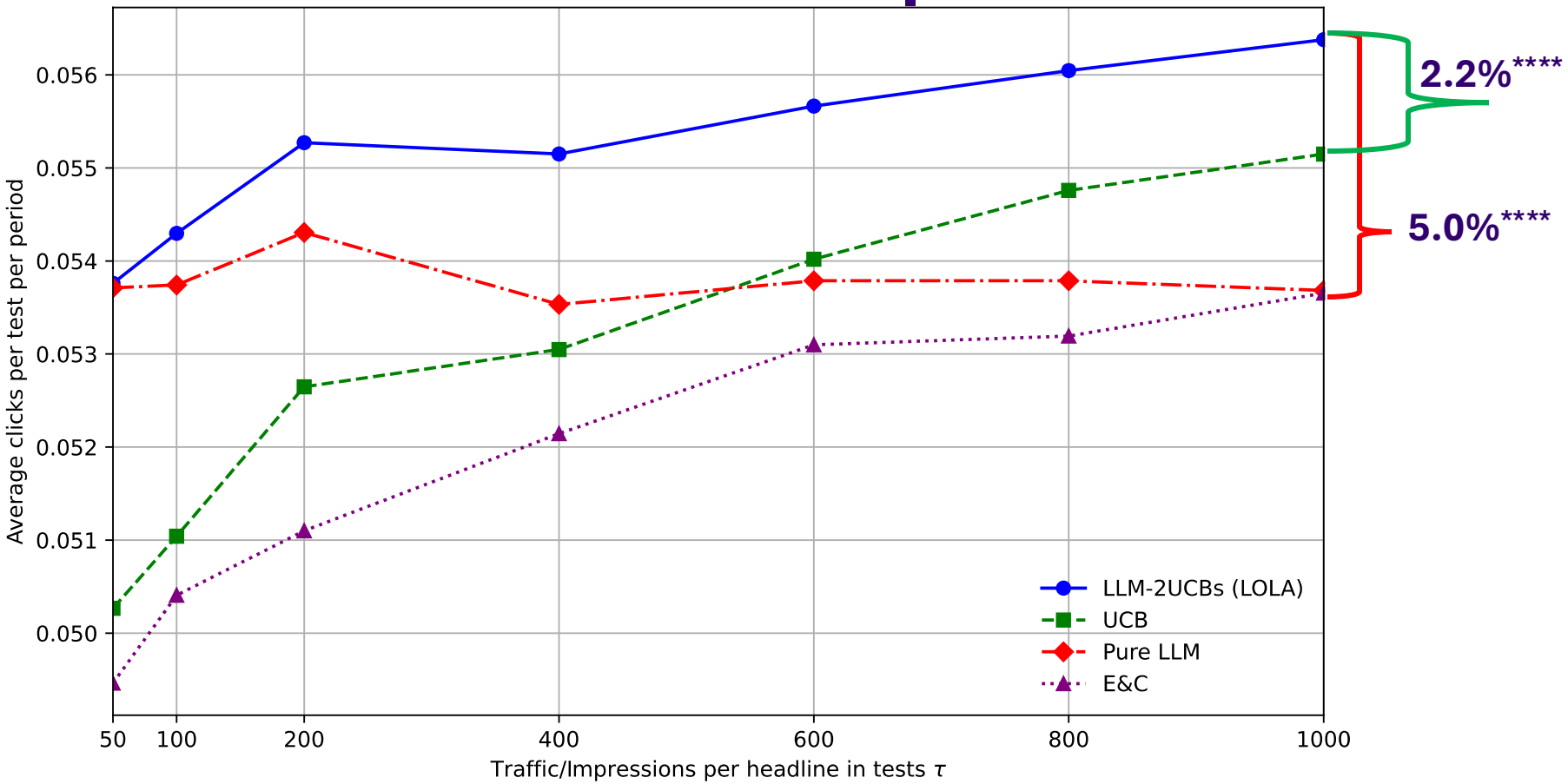
Experiments and Benchmarks

We use all Upworthy data for headline tests, with18k tests and 75k headlines.

We compare:

- **LOLA:** LLM + UCB (our approach)
- **UCB:** Pure UCB without LLM
- **Pure LLM:** Pure LLM (LoRA fine-tuning) without UCB
- **E&C:** Explore and Commit, test and roll (Upworthy's practice)

Performance Comparison



X-axis: scale of the time horizon. Average impressions per headline
Y-axis: clicks per test per period.

- LOLA performs the best.
- The gap between LOLA and UCB gets smaller under larger T.
- Pure LLM performs consistently, but worse than A/B test under larger T.

Advantages of LOLA

- Plug and play nature makes it widely applicable
  - Step 1 is compatible with any LLM-based method to obtain CTR and step 2 is compatible with any standard bandit algorithm (UCB/TS)
  - Easily integrated into existing online experimentation systems
- Can be used with other outcome variables
  - E.g., Time spent on article
- Optimize a different goal
  - Best arm identification instead of regret minimization
- Can be used with AI generated headlines
  - Can feed the results back into LLMs to generate headlines over a longer horizon
- Easy to personalize
  - Can be extended to include user and context features

Contributions and Conclusions

- **Methodological:** A novel framework for content experimentation in the era of LLMs.
  - A general framework that can accommodate a wide range of LLMs and bandits
- **Empirical:** We empirically show that LOLA performs better than the standard A/B test, and bandits.
  - Particularly valuable in news industry and limited traffic settings
- **Managerial:** LOLA is easy to use and only requires small modification to standard experimentation platforms.
  - Applicable to other settings such as digital advertising, email marketing, and website design

Thank You!

Email: hemay@uw.edu

Code: https://github.com/DDDOH/LLM_News