

Kernel-Based Representation Learning for Experimentation with LLM-Generated Treatments

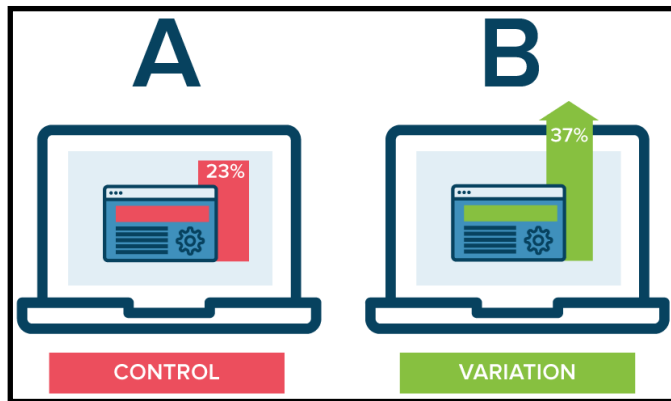
Lei Shi

Joint work with: David Arbour, Raghav Addanki, Ritwik Sinha, Avi Feller

2024 Summer @ Adobe Research

Presenting @ Casual Causal, 02/06/2025

Motivation

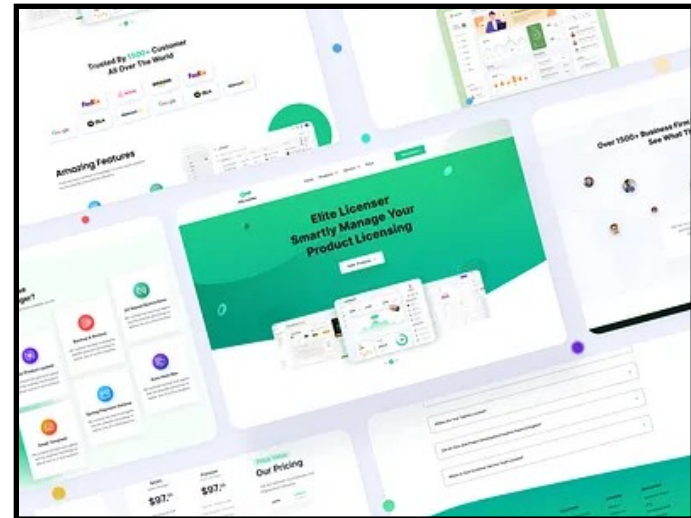


Classical A/B testing:

- markets design treatment & control variants (website, advertisement, etc.)
- test them on different traffic of users
- **Easy testing, hard designing!**

LLMs are the game-changer:

- markets generate a large number of variants with one simple prompt
- Test all of them on users
- **Easy designing, hard testing!**



Motivation

- Ex: Design a headline for casual causal website with GPT
- Generate ten headlines with one click!
- Can generate much more...
- But how to evaluate them?
(Estimation)
- And how to efficiently collect data to decide the best arm?
(Design)

I try to design a headline for one web page, which is for promoting a group in Berkeley working on causal inference. The group is casual causal. Give me ten headline for the webpage!



Here are ten headline ideas for the **Casual Causal** group at Berkeley, promoting their work on causal inference:

1. "Casual About the Name, Serious About Causality – Join Us at Berkeley!"
2. "Causal Inference, Without the Complications – Welcome to Casual Causal"
3. "Making Causal Inference Approachable – The Casual Causal Way"
4. "Where Rigor Meets Relaxation – Exploring Causality at Berkeley"
5. "Decoding Causal Effects, One Experiment at a Time – Join Casual Causal"
6. "Serious Causal Science, No Frills – Meet the Casual Causal Group"
7. "Causal Thinking, Without the Jargon – A Friendly Approach to Inference"
8. "From Correlation to Causation – Let's Make It Casual"
9. "Challenging Assumptions, Testing H_0 Hypotheses – The Casual Causal Mindset"
10. "Join Berkeley's Chillest, Sharpest Causal Inference Community"

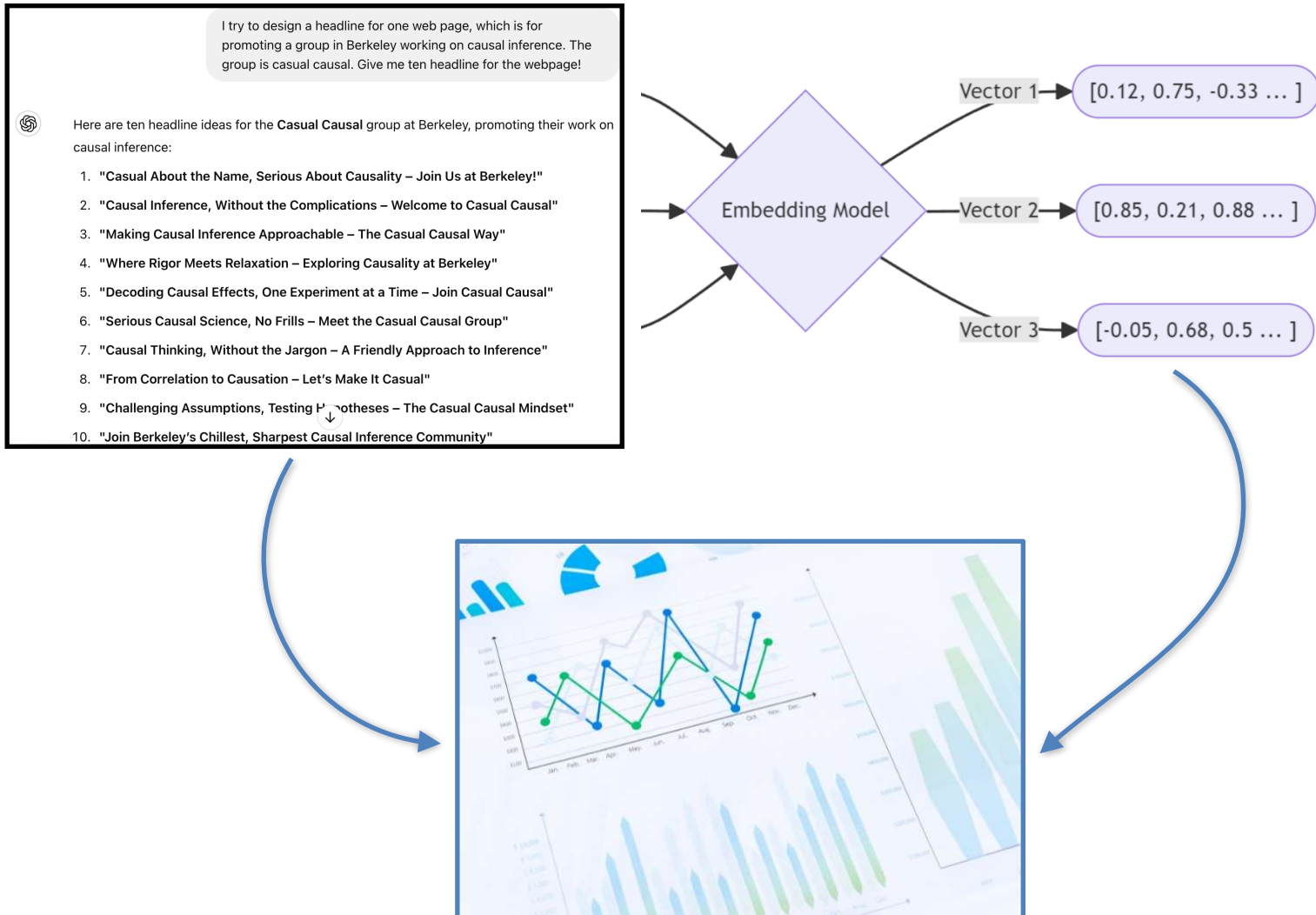
Difficulty of Classical Thinking

- Ten headlines as ten discrete arms...
- Textbook-style experimental design + causal inference, with multiple treatments levels
- Problem solved! ... or no?
 - Treatments are special: text (or even other modality) as treatment
 - Inference is underpowered with a large number of arms
 - New treatments may come

Our idea: utilize the semantic information

- **Semantic information of the treatment matters!**
 - Treatments are special: text as treatment (semantic information is encoded)
 - Inference is underpowered with a large number of arms (semantic similarity to pool information)
 - New treatments may come (semantic information for predicting the behavior)
- **Embeddings:** numerical representation of the semantic information
 - More of a CS jargon; for us maybe think of it as “covariates for treatments”

Our idea: utilize the semantic information



Problem description

- A causal inference framework: consider the following structural equations:

$$y(z) = f(z, x) + \epsilon, \quad z \in \mathcal{Z}, \quad x \in \mathcal{X}.$$

- The CATE is defined as:

$$\tau(z, x) = f(z, x) - f(z_0, x).$$

- the difficulty of treating \mathcal{Z} as discrete: too many arms + no semantic information pooling!
- Instead we represent \mathcal{Z} as the embedding space
 - To note: we are focusing on estimation instead of identification issues here!

Warm up: learning with treatment embeddings

- Consider a simple first step: $f(z, x) = f(z)$
- **Kernel methods are standard tools for measuring similarity with continuous parametrization!**
- Examples:

Linear kernel:

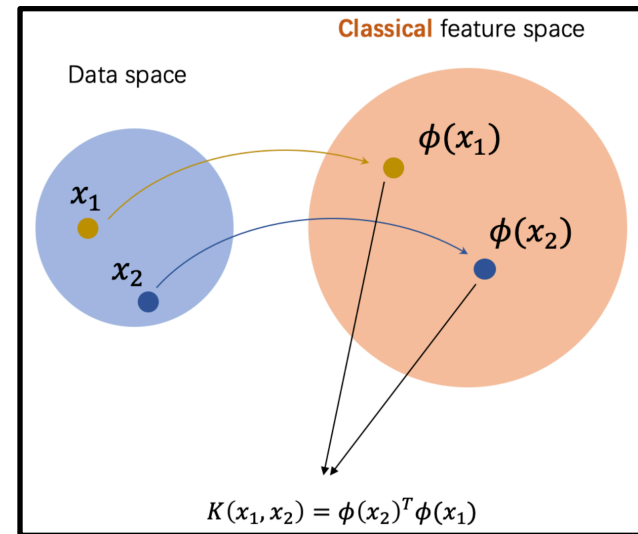
$$K(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \mathbf{x}^{(i)T} \mathbf{x}^{(j)}$$

Polynomial kernel:

$$K(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \left(1 + \mathbf{x}^{(i)T} \mathbf{x}^{(j)}\right)^d$$

Radial basis functions (RBF)

$$K(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \exp\left(-\frac{(\mathbf{x}^{(i)} - \mathbf{x}^{(j)})^2}{\sigma^2}\right)$$



Warm up: learning with treatment embeddings

- Start simple: how about starting from a linear similarity case? - let's run ridge! (**estimate**)

$$\hat{f}(z) = \phi(z)^\top \hat{\beta}, \quad \hat{\beta} = \frac{1}{2n} \sum_{i=1}^n \{y_i - \phi(z_i)^\top \beta\}^2 + \lambda_n \|\beta\|_2^2.$$

- Maximizing \hat{f} to identify the best arm
- Make the algorithm adaptive (**design**): put a Gaussian prior on β and update the posterior distribution of y
(Fancier name: Thompson Sampling!)

Warm up: learning with treatment embeddings

- Can we move to more general nonlinear models?
- Kernel ridge regression with embeddings:

$$\hat{f} = \arg \min_{f \in \mathbb{H}_{\mathcal{K}}} \frac{1}{2n} \sum_{i=1}^n (y_i - f(z_i))^2 + \lambda_n \|f\|_{\mathbb{H}_{\mathcal{K}}}.$$

- Results in an interpolator:

$$\hat{f}(z) = \sum_{i=1}^n \hat{\alpha}_i \mathcal{K}(z, z_i).$$

- Then we can maximize \hat{f} !
- To make it adaptive: put a prior on $f(z)$ and update posterior (**Fancy name: Gaussian Process, Bayes Opt**)

$$\begin{bmatrix} f(x_1) \\ \vdots \\ f(x_m) \end{bmatrix} \sim \mathcal{N} \left(\begin{bmatrix} m(x_1) \\ \vdots \\ m(x_m) \end{bmatrix}, \begin{bmatrix} k(x_1, x_1) & \cdots & k(x_1, x_m) \\ \vdots & \ddots & \vdots \\ k(x_m, x_1) & \cdots & k(x_m, x_m) \end{bmatrix} \right).$$

Our goal: Incorporating customer covariates

- Hope to extend the idea to incorporate covariates
- For n customers, the expected potential outcome matrix is:

$$\begin{pmatrix} f(z_1, x_1) & \dots & f(z_1, x_n) \\ \vdots & \ddots & \vdots \\ f(z_n, x_1) & \dots & f(z_n, x_n) \end{pmatrix}$$

- Ideal world: Multi-task Gaussian process
- Reality: Many missing pairs!

Incorporating covariates

- Let's Revisit estimation of CATE: R-learner by Nie & Wager

- For binary treatment: partial linear model

$$y = f(z_0, x) + \mathbf{1}\{z = z_1\} \cdot \tau(x) + \epsilon.$$

- a decomposition of treatment effects
- Conditioning on covariates:

$$y = m(x) + (\mathbf{1}\{z = z_1\} - e(x)) \cdot \tau(x) + \epsilon.$$

- Learn the function $\tau(x)$

Incorporating covariates

- Mimicking the idea
 - For general treatment: partial linear model

$$y = f(z_0, x) + \mathbf{g}(z)^\top \mathbf{h}(x) + \epsilon.$$

- A decomposition of treatment effects into low-dimensional summary with unknown **representations** \mathbf{g}, \mathbf{h} !
- Conditioning on covariates:

$$m(x) = f(z_0, x) + \bar{\mathbf{g}}(x)^\top \mathbf{h}(x).$$

$$y = m(x) + (\mathbf{g}(z) - \bar{\mathbf{g}}(x))^\top \mathbf{h}(x) + \epsilon.$$

- **Need to learn both $\mathbf{g}(z)$ and $\mathbf{h}(x)$!**

Incorporating covariates

- Let's simplify the problem: Experimental data where propensity score does not depend on x
 - No dependence of $\bar{g}(x)$ on x :

$$y = m(x) + (\mathbf{g}(z) - \bar{\mathbf{g}})^\top \mathbf{h}(x) + \epsilon.$$

- We can assume $g \in \mathbb{H}(\mathcal{K}_g)$ and $h \in \mathbb{H}(\mathcal{K}_h)$
- A natural extension of the kernel ridge:

$$(\hat{\mathbf{g}}, \hat{\mathbf{h}}) = \arg \min_{\{g_l, h_l\}_{l=1}^r} \frac{1}{2n} \sum_{i=1}^n \left\{ y_i - \sum_{l=1}^r g_l(z_i) h_l(x_i) \right\}^2 + \lambda_n \sum_{l=1}^r (\|g_l\|_{\mathcal{K}_g}^2 + \|h_l\|_{\mathcal{K}_h}^2).$$

**Double Kernel
Representation
Learning**

Double Kernel Representation Learning

- How to solve this from data?
- **A representer theorem:**
 - Representing the RKHS elements by function bases constructed from data
 - Two kernels and two functions:

$$U = [U_1^R, \dots, U_n^R]^\top \in \mathbb{R}^{n \times r},$$
$$V = [V_1^R, \dots, V_n^R]^\top \in \mathbb{R}^{n \times r},$$

$$\hat{g}(z) = \sum_{i \in [n]} U_i^R \mathcal{K}_g(z, z_i), \quad \hat{h}(x) = \sum_{i \in [n]} V_i^R \mathcal{K}_h(x, x_i),$$

$$f(z, x) = \sum_{i \in [n]} \sum_{j \in [n]} \langle U_i^R, V_j^R \rangle \mathcal{K}_g(z, z_i) \mathcal{K}_h(x, x_j)$$
$$= \mathcal{K}_g(z, \mathbf{z}_{1:n}) \Theta \mathcal{K}_h(x, \mathbf{x}_{1:n})^\top,$$

- U, V can be solved from an alternating algorithm
- Statistical convergence under a fixed-base setting

An Adaptive Experimentation Strategy

- Explore-then-commit for adaptive treatment allocation
 - Initial T_e stages (exploration) - randomly sample treatment and users to learn the outcome function
 - Following T_c stages (exploitation) - randomly draw a user, and assign her to the best estimated treatment
- A sublinear gap between realized outcomes and optimal (regret)



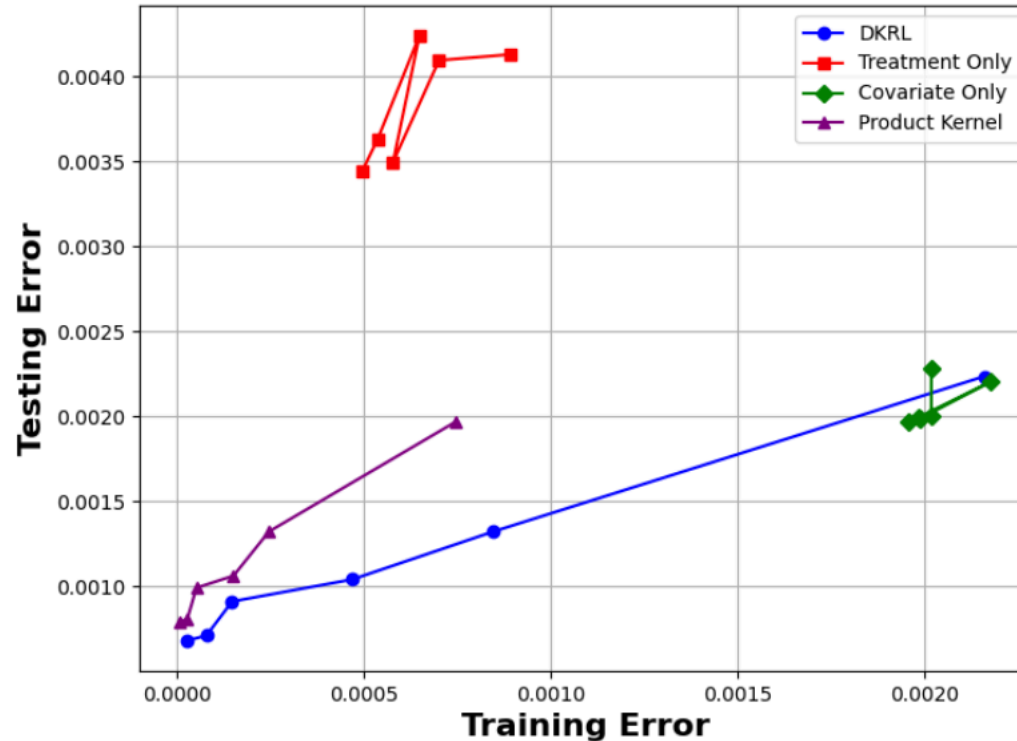
Synthetic Simulation: Upworthy Research Archive

- **Upworthy Research Archive:** an open dataset of thousands of A/B tests of headlines conducted by Upworthy from January 2013 to April 2015.
- Contains headlines of the ads on the website and customer reactions
- We generate synthetic data from Upworthy:
 - Get embeddings of the headlines from sentence transformer
 - Simulate covariates from multivariate gaussian
 - Generate outcome from a bilinear low rank model:

$$f(z, x) = z^{\top} \Theta^* x$$

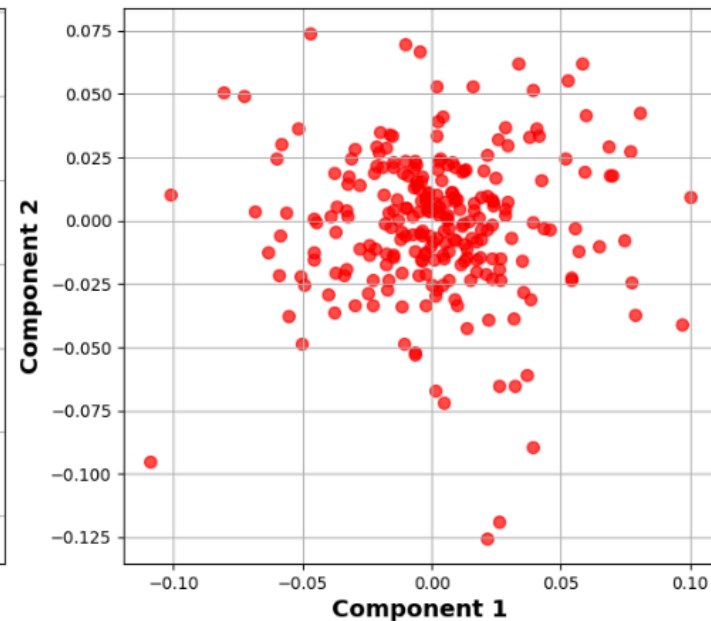
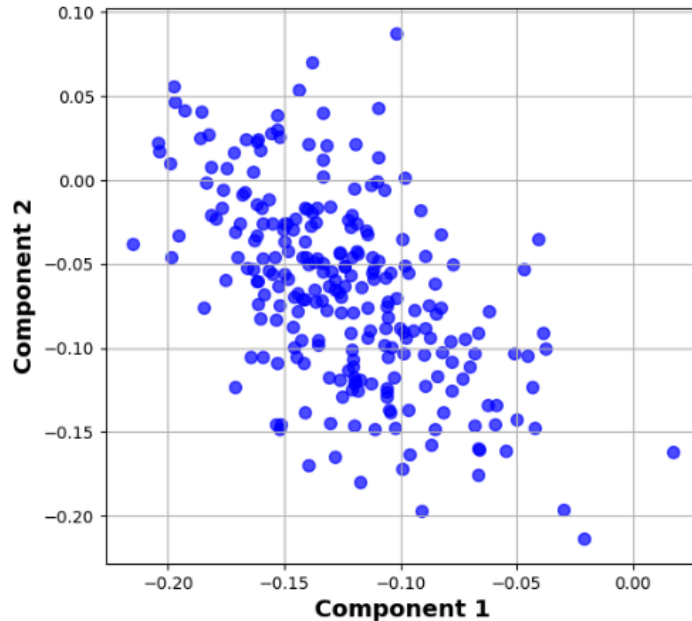
Synthetic Simulation: Upworthy Research Archive

- Compare four methods: (i) DKRL; (ii) treatment only; (iii) covariate only; (iv) Product kernel



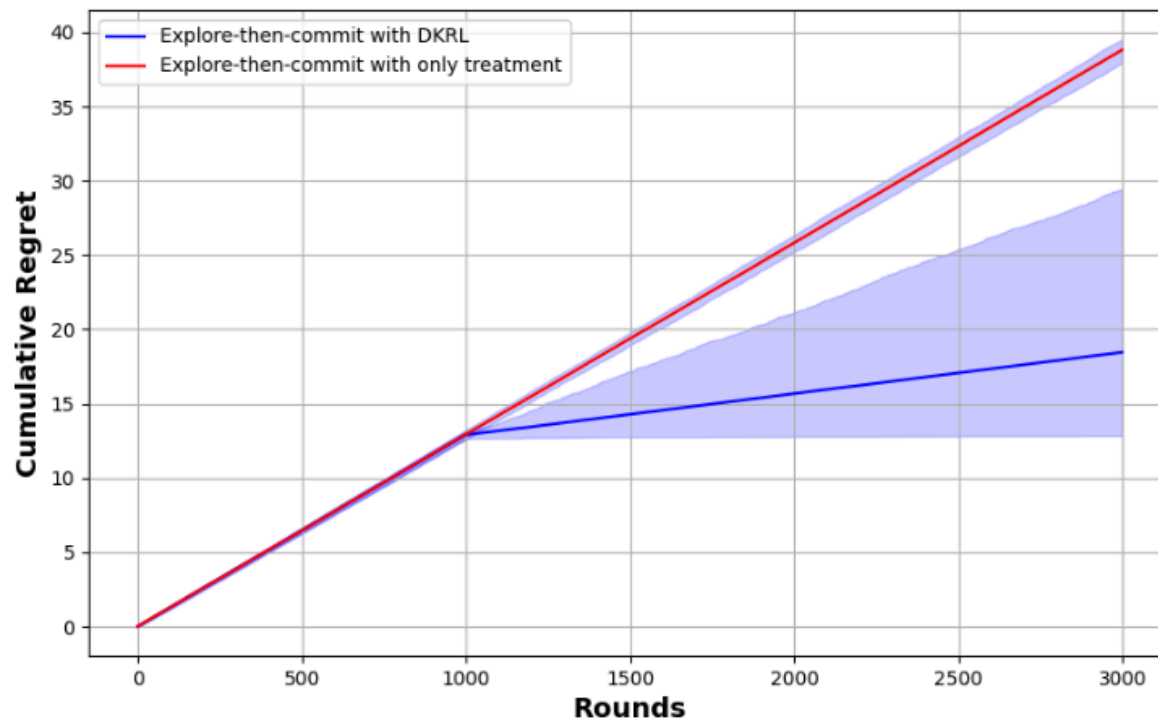
Synthetic Simulation: Upworthy Research Archive

- DKRL learns the low dimensional representation of the data



Synthetic Simulation: Upworthy Research Archive

- Explore-then-commit with DKRL has a sublinear regret



Conclusion and Future Directions

- DKRL improves statistical efficiency in the design and analysis of LLM-generated treatment experiments.
- Possible extension: Handling delayed responses, privacy concerns, and network interference.

APPENDIX

Connection with classical methods

- The fixed-base version: low rank matrix factorization
- The Gaussian process version: Kernelized probability matrix factorization
-