Aligning Language Models with Observational Data: Opportunities and Risks from a Causal Perspective

Erfan Loghmani *
Foster School of Business
University of Washington
Seattle, WA 98195
loghmani@uw.edu

Abstract

Large language models are being widely used across industries to generate text that contributes directly to key performance metrics, such as medication adherence in patient messaging and conversion rates in content generation. Pretrained models, however, often fall short when it comes to aligning with human preferences or optimizing for business objectives. As a result, fine-tuning with good-quality labeled data is essential to guide models to generate content that achieves better results. Controlled experiments, like A/B tests, can provide such data, but they are often expensive and come with significant engineering, logistical, and ethical challenges. Meanwhile, companies have access to a vast amount of historical (observational) data that remains underutilized. In this work, we study the challenges and opportunities of fine-tuning LLMs using observational data. We show that while observational outcomes can provide valuable supervision, directly fine-tuning models on such data can lead them to learn spurious correlations. We present empirical evidence of this issue using various real-world datasets and propose DECONFOUNDLM, a method that explicitly removes the effect of known confounders from reward signals. In simulation experiments, DECONFOUNDLM more accurately recovers causal relationships and mitigates failure modes of methods that assume counterfactual invariance, achieving over 16% higher objective score than ODIN and other baselines, when entangled confounding is present. Please refer to the project page for code and related resources.

1 Introduction

Large language models (LLMs) can be powerful tools for creating content that affects user behavior and supports business goals. From enhancing user engagement to increasing purchase likelihood, companies often aim to generate content that delivers measurable outcomes. Prior research has shown that pretrained LLMs can perform well in certain tasks, such as generating creative product ideas [1] and predicting the likelihood of purchases [2]. However, these models often struggle to capture human preferences and directly optimize for business outcomes [3, 4], emphasizing the need for fine-tuning with labeled data to causally guide the models towards desired business outcomes. Yet obtaining the right kind of labeled data to support this alignment is difficult; human-labeled data and surveys can introduce bias due to artificial contexts [5], and randomized experiments are often infeasible due to logistical and opportunity costs [6]. This paper explores how to bridge this gap using an abundant but underutilized source of supervision available to firms: historical observational data.

*The author would like to thank Mahtab Bigverdi, Shirsho Biswas, Ali Goli, Lalit Jain, Max Kleiman-Weiner, Jacques Lawarrée, Amin Sayedi, Amandeep Singh, Hema Yoganarasimhan, as well as the participants of the University of Washington RAIVN Lab seminar series and the Twenty-Sixth ACM Conference on Economics and Computation (EC'25), for their valuable feedback and thoughtful suggestions.

Accepted to the NeurIPS 2025 Workshop on CauScien: Uncovering Causality in Science.

Consider a news website that aims to improve the click-through rates (CTR) of news headlines. While they may not have the capacity to run controlled experiments, they may track how users respond to different headlines over time, which could be used for fine-tuning. However, directly fine-tuning on this data could be challenging because external factors such as time trends may influence both the content and the outcome. Furthermore, while running controlled experiments may already be impractical for a news website, the difficulty is amplified in healthcare settings, where fairness and ethical concerns further restrict experimentation and increase the importance of leveraging available historical data. In this paper, we examine both the opportunities and risks associated with using observational data, and we propose a novel method that corrects for confounding effects in the fine-tuning process.

Fine-tuned LLMs have been used in many domains from electronic health records [7] to astronomical data [8] and social-science corpora, however most work does not discuss causal challenges or the pitfalls of learning from historical data. In business settings, methods such as Ye et al. [4] and Angelopoulos et al. [9] rely on experimental supervision, leaving the potential of firms' extensive historical logs underexplored. On the causal side, computer science work has documented biases in preference data used for fine-tuning (e.g., length bias and sycophancy) and proposed methods, such as ODIN [10], Wang et al. [11], and Srivastava et al. [12] that rely on the counterfactual invariance assumption. This assumption forces the reward estimates not to change when confounding attributes are changed. While counterfactual invariance may be plausible for human-rated LLM outputs, it is often violated in real-world applications where different variables can affect outcomes. In contrast, we propose a method to use abundant observational data and correct for observed confounders without assuming counterfactual invariance. A more detailed discussion on the related work can be found in Appendix A.

Our results in this paper show:

- **Risks of observational signals.** Using *StackExchange*, we show that naive fine-tuning on historical interactions can lead models to internalize spurious correlations.
- **Value of observational signals.** Using *Upworthy*, we show that observational data can provide valuable signals. We also highlight the role of regularization to suppress the confounding effects when using historical data.
- **Confounder correction.** We propose DECONFOUNDLM, a fine-tuning method that removes the influence of observed confounders from the reward signal. Our results show that this approach consistently improves model behavior, enabling it to focus on causally relevant attributes rather than superficial artifacts.

2 Observational Data: Pitfalls and Potential

In this section, we examine both the pitfalls and potential of learning from historical observational data, relying on experiments using StackExchange and Upworthy data.

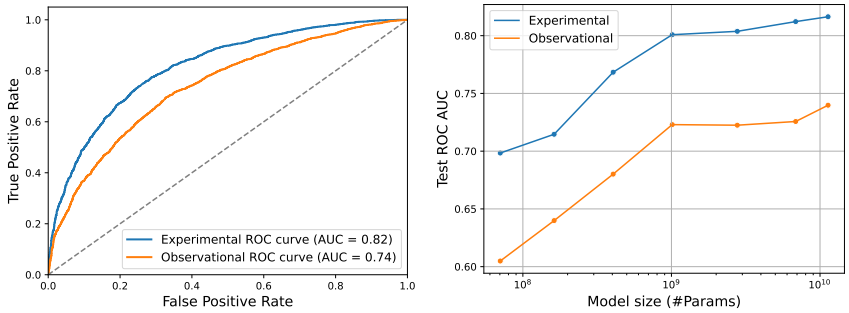
Pitfall: Internalizing spurious correlation. We illustrate how confounding in historical data can mis-specify rewards when fine-tuning LMs. Using Academia Stack Exchange, we mimic Askell et al. [13] by treating user scores as preferences. In this dataset, engagement varies by weekday, specifically, we see a higher activity earlier in the week (Figure 2). Because of this pattern, scores partly reflect exposure rather than quality. To make this spurious signal explicit, we prepend a "Happy Monday!" marker to Monday answers, then construct preference pairs where the higher-scored answer is treated as preferred. Evaluated on 3,000 held-out questions, models trained on these preferences learn the weekday cue: compared to SFT, DPO amplifies the artifact, generating "Happy" 21.2% vs. 13.7% and "Monday" 11.8% vs. 9.1% of the time (both increases statistically significant). Full description and details of this experiment are presented in Appendix C.

Potential. In the previous part, we showed that historical data can induce spurious correlations. Here we ask: *What is its potential value?* When a firm lacks experimental data, can logs of content and observed performance still improve future predictions? This cannot be answered in purely observational settings because outcomes are not causally attributable to the content. We therefore use the Upworthy dataset [14], which provides CTRs from controlled A/B tests. To simulate observational

access, we retain a single headline package and its CTR from each test and discard the alternative; dataset statistics and preprocessing are discussed in Appendix D.

To compare learning from experimental vs. observational data, we fine-tune LLMs by adding a small head to the final embeddings: (i) a pairwise classification head with logistic loss to predict the higher-CTR headline (experimental), and (ii) a regression head with MSE to predict observed CTR (observational). In both settings, we use L_2 regularization and tune λ on a validation set.

We evaluate all models on the same held-out set of headline pairs, where each pair comes from an A/B test with a known preferred headline. The evaluation objective is to assess whether the model correctly ranks the preferred headline higher. To do this, we compute the ROC AUC (Area Under the Receiver Operating Characteristic Curve), which reflects the model's ability to distinguish between better and worse-performing headlines. We use the Pythia suite of open-weight language models for all experiments [15]. Figure 1a shows the ROC AUC results for the Pythia-12B model. Training on the experimental dataset yields an AUC of 0.82, whereas the observational dataset produces a lower, but still above-chance, AUC of 0.74. This result is encouraging: it suggests that historical data, even without experimental variation and with only about 26% of the training packages, can still provide meaningful signals for preference learning. The performance gap also highlights the value of randomized feedback; exposure to counterfactual comparisons enables better generalization and more reliable preference estimation. Figure 1b further shows AUC improves with model size in both settings, yet the experimental–observational gap remains, underscoring the value of randomized feedback when available. Our experiments also show important observations on the role of regularization when using historical data and its effect on confounding; however, to save space, we defer these results and details to Appendix D.



(a) Pythia-12B comparison: experimental vs. observational.

(b) ROC AUC across model sizes.

Figure 1: ROC on held-out Upworthy headline pairs. (a) ROC curves for models trained with experimental data outperform those trained on observational data. (b) Larger models yield better results in both settings, but the performance gap persists.

3 A causal framework for observational fine-tuning

We now introduce a formal framework to analyze the effect of confounding in observational fine-tuning. Suppose we have access to historical interaction data $\mathcal{D} = \{(\boldsymbol{X}_i, y_i)\}_{i \in \mathcal{I}}$, where $\boldsymbol{X}_i$ denotes the context vector for interaction i and y_i is the associated outcome (e.g., click-through rate or reservation rate). We assume that $\boldsymbol{X}_i$ can be decomposed into a textual decision variable T_i (e.g., a generated title or headline) and auxiliary features $\tilde{\boldsymbol{X}}_i$, such that: $y_i = f(T_i, \tilde{\boldsymbol{X}}_i) + \epsilon_i$, where f is the unknown outcome function and ϵ_i is zero-mean noise. The goal is to train a generative model $G : \tilde{\boldsymbol{X}} \to T$ that produces high-reward textual actions for new inputs. To capture confounding, we assume that the outcome function can be decomposed as:

$$y_i = g(T_i, \tilde{\boldsymbol{F}}_i) + h(\boldsymbol{C}_i) + \epsilon_i, \tag{1}$$

where $\boldsymbol{F}_i$ is the set of observed features including T_i, and $\boldsymbol{C}_i$ represents observed confounders that influence both the action and the outcome. The function g captures the causal effect of the textual

action and other features, while h captures the contribution of confounders. Crucially, if g and h are entangled, estimating them independently may result in biased models and reward misspecification.

Proposed method: DECONFOUNDLM Our proposed approach, Deconfounded Language Model Fine-Tuning (DECONFOUNDLM), involves first identifying and modeling the effect of confounders, and then explicitly removing their contribution from the observed outcomes. This allows the model to learn the causal impact of the textual input and other features, without being influenced by confounding effects. In our experiments, we apply an instrumental variable strategy to estimate the confounding component. However, the framework is flexible: other methods such as Double Machine Learning [16] or Adversarial GMMs [17] can be used, provided the researcher is mindful of the assumptions of the methods. We further discuss the potential impacts and limitations in Appendix F.

In this paper, we demonstrate the effectiveness of using instrumental variable (IV) estimation within our DECONFOUNDLM framework. While finding ideal instruments may be challenging in many business applications, suitable instruments do exist and have been studied in economics and social sciences. See Imbens [18] for a review in econometrics, Sovey and Green [19] in political science, and Wu et al. [20] for a recent general survey of the use of IV in causal inference and machine learning.

3.1 Simulation Experiments

In this section, we turn to simulation experiments for a controlled evaluation of the proposed DECONFOUNDLM method. We base our simulation experiments on the MIND dataset [21], which contains over 160,000 English-language news articles with both titles and full text. We treat the article body as the input context and aim to generate a headline T_i that maximizes a synthetic performance score y_i, interpreted as a proxy for engagement or click-through rate. To simulate realistic challenges in observational fine-tuning, we design two scenarios, *Orthogonal confounding* and *Entangled confounding*, where the observed outcome y_i depends on both the textual quality and a confounding variable p_i representing topic popularity (e.g., how much fan interest a team garners). We use the headline sentiment $s(T_i) = g(T_i, \tilde{\boldsymbol{F}}_i)$ as a measure of quality, as it is interpretable and easily measured.

Across both scenarios, we model the outcome as:

$$y_i = s(T_i) + 0.1\, p_i + \nu_i, \tag{2}$$

where p_i is the confounder, and $\nu_i \sim \mathcal{N}(0, 0.1)$ represents observational noise. We vary how p_i is constructed across two settings:

- *Orthogonal confounding.* The confounder p_i is independent of the sentiment $s(T_i)$, making its effect easier to isolate and remove. Specifically:

$$p_i = \mathbb{1}(\text{title mentions West Coast team}) + 2 \cdot \mathbb{1}(\text{Central team}) + 3 \cdot \mathbb{1}(\text{East Coast team}) + \epsilon_i, \tag{3}$$

  where $\epsilon_i \sim \mathcal{N}(0, 0.5)$. This reflects a hypothetical bias where East Coast teams are generally more popular and draw higher engagement regardless of the title's quality.

- *Entangled confounding.* Here, popularity p_i is correlated with the sentiment of the news abstract, mimicking a setting where emotional salience of the topic and engagement co-vary. For instance, sad events may draw more audience to the platform and lead to increased engagement. We model this with:

$$\begin{aligned} p_i = {} & \mathbb{1}(\text{title mentions West Coast team}) + 2 \cdot \mathbb{1}(\text{Central team}) + 3 \cdot \mathbb{1}(\text{East Coast team}) \\ & - 10.5 \cdot s(\text{abstract}) + \epsilon_i. \end{aligned} \tag{4}$$

IV justification. In both scenarios, the sentiment of the title $s(T_i)$ and the popularity measure p_i affect engagement Y_i. Omitting popularity from the analysis induces bias in the estimated effect of headline sentiment on Y_i. The same issue happens when we want to estimate the effect of p_i on Y_i without controlling for the text, specifically in the Entangled confounding setting. Instrumental variables (IV) address this problem by exploiting variation that affects the outcome only through its

impact on popularity. In our case, team mentions satisfy this requirement: they generate exogenous shocks to popularity (e.g., attention surges around specific sporting events) but do not otherwise alter the causal path from headline sentiment to engagement.

The IV procedure estimates and removes the contribution of popularity, leaving only the component of engagement that is causally attributable to headline sentiment. This illustrates the classical conditions under which IV estimation is effective: (i) *relevance*, since team mentions are strongly correlated with popularity, and (ii) *exclusion*, since they affect engagement only through popularity. When these conditions are satisfied, IV recovers the true causal effect of headline sentiment even in the presence of confounding.

Comparative Methods. We evaluate seven approaches: (1) a base pre-trained model, (2) supervised fine-tuning (SFT), (3) RL with access to ground-truth sentiment (which serves as a baseline) (4) RL using observed performance without controlling for confounders, (5) RL models that incorporate popularity either as input text or as a scalar feature in the final layer, (6) ODIN [10] as an example of a method that relies on counterfactual invariance, and (7) our proposed method DECONFOUNDLM-IV, which estimates and removes the confounder effect using an instrumental variable.

Results. We evaluate all models' generations after the RL step on a held-out set of 3,000 news articles. In the *Orthogonal* setting, the model trained on observed performance (without accounting for confounding) is able to improve headline sentiment, indicating that it learns part of the true signal. However, it also shows a marked increase in the frequency of team name mentions, suggesting reliance on popularity cues. Incorporating popularity information, either via text prompts or as an input feature, reduces this effect. Among all methods, DECONFOUNDLM-IV more closely matches the sentiment gains of the true-reward model while not generating unnecessary references to team names caused by the confounding variable.

The *Entangled* case presents a more challenging scenario. Here, the naive model trained on observed performance fails to improve sentiment and heavily generates team names. While models that include popularity in the input text or final layer performed well in the *orthogonal* setting, they struggle to recover the sentiment-performance relationship in this setting. ODIN also achieves a similar performance in terms of sentiments of the generated headlines, but generates many more team names. In contrast, DECONFOUNDLM-IV demonstrates strong robustness. It successfully suppresses the influence of the confounder and generates headlines with **16%** higher sentiment scores compared to the next highest sentiment scores. Full experimental details and results are provided in Appendix E.

4 Conclusion and discussion

Our findings suggest that using historical data to fine-tune language models can be a double-edged sword: while it provides valuable information without the need for experimentation, it could also introduce the risk of learning from confounded outcomes. Through both real-world and synthetic experiments, we show that models trained on observational data may internalize spurious correlations that are not causally linked to content quality. To mitigate this, we introduce DECONFOUNDLM, a method that explicitly adjusts for observed confounders in the fine-tuning process. By separating confounding influences from the outcome signal, our approach enables more causally grounded learning without relying on the counterfactual invariance assumption that is often used in prior work. Across multiple settings, we find that DECONFOUNDLM improves fine-tuning outcomes and better captures the true effects of textual inputs. Finally, while our primary focus is performance and causal inference, we note that confounding can also introduce fairness concerns. If unaddressed, it may lead models to replicate or amplify structural biases in the data. We view causal deconfounding as a promising direction for aligning language models not only with user preferences but also with broader values of equity and accountability.

References

[1] Noah Castelo, Zsolt Katona, Peiyao Li, and Miklos Sarvary. How ai outperforms humans at creative idea generation. *Available at SSRN 4751779*, 2024.

[2] Neeraj Arora, Ishita Chakraborty, and Yohei Nishimura. Ai–human hybrids for marketing research: Leveraging large language models (llms) as collaborators. *Journal of Marketing*, 89 (2):43–70, 2025.

[3] Ali Goli and Amandeep Singh. Frontiers: Can large language models capture human preferences? *Marketing Science*, 43(4):709–722, 2024.

[4] Zikun Ye, Hema Yoganarasimhan, and Yufeng Zheng. Lola: Llm-assisted online learning algorithm for content experiments. *arXiv preprint arXiv:2406.02611*, 2024.

[5] Min-Hsuan Yeh, Leitian Tao, Jeffrey Wang, Xuefeng Du, and Yixuan Li. How reliable is human feedback for aligning large language models? *arXiv preprint arXiv:2410.01957*, 2024.

[6] Federico Quin, Danny Weyns, Matthias Galster, and Camila Costa Silva. A/b testing: A systematic literature review. *Journal of Systems and Software*, page 112011, 2024.

[7] Zhenbang Wu, Anant Dadu, Mike Nalls, Faraz Faghri, and Jimeng Sun. Instruction tuning large language models to understand electronic health records. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 54772–54786. Curran Associates, Inc., 2024. URL `https://proceedings.neurips.cc/paper_files/paper/2024/file/62986e0a78780fe5f17b495aeded5bab-Paper-Datasets_and_Benchmarks_Track.pdf`.

[8] Yu Wang, Shu-Rui Zhang, Aidin Momtaz, Rahim Moradi, Fatemeh Rastegarnia, Narek Sahakyan, Soroush Shakeri, and Liang Li. Can ai understand our universe? test of fine-tuning gpt by astrophysical data. *arXiv preprint arXiv:2404.10019*, 2024.

[9] Panagiotis Angelopoulos, Kevin Lee, and Sanjog Misra. Causal alignment: Augmenting language models with a/b tests. *Available at SSRN*, 2024.

[10] Lichang Chen, Chen Zhu, Davit Soselia, Jiuhai Chen, Tianyi Zhou, Tom Goldstein, Heng Huang, Mohammad Shoeybi, and Bryan Catanzaro. Odin: Disentangled reward mitigates hacking in rlhf. *arXiv preprint arXiv:2402.07319*, 2024.

[11] Chaoqi Wang, Zhuokai Zhao, Yibo Jiang, Zhaorun Chen, Chen Zhu, Yuxin Chen, Jiayi Liu, Lizhu Zhang, Xiangjun Fan, Hao Ma, et al. Beyond reward hacking: Causal rewards for large language model alignment. *arXiv preprint arXiv:2501.09620*, 2025.

[12] Pragya Srivastava, Harman Singh, Rahul Madhavan, Gandharv Patil, Sravanti Addepalli, Arun Suggala, Rengarajan Aravamudhan, Soumya Sharma, Anirban Laha, Aravindan Raghuveer, et al. Robust reward modeling via causal rubrics. *arXiv preprint arXiv:2506.16507*, 2025.

[13] Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas Joseph, Ben Mann, Nova DasSarma, et al. A general language assistant as a laboratory for alignment. *arXiv preprint arXiv:2112.00861*, 2021.

[14] J Nathan Matias, Kevin Munger, Marianne Aubin Le Quere, and Charles Ebersole. The upworthy research archive, a time series of 32,487 experiments in us media. *Scientific Data*, 8 (1):195, 2021.

[15] Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR, 2023.

[16] Victor Chernozhukov, Denis Chetverikov, Mert Demirer, Esther Duflo, Christian Hansen, Whitney Newey, and James Robins. Double/debiased machine learning for treatment and structural parameters. *The Econometrics Journal*, 21(1):C1–C68, 2018.

[17] Nishanth Dikkala, Greg Lewis, Lester Mackey, and Vasilis Syrgkanis. Minimax estimation of conditional moment models. *Advances in Neural Information Processing Systems*, 33: 12248–12262, 2020.

[18] Guido Imbens. Instrumental variables: an econometrician's perspective. Technical report, National Bureau of Economic Research, 2014.

[19] Allison J Sovey and Donald P Green. Instrumental variables estimation in political science: A readers' guide. *American Journal of Political Science*, 55(1):188–200, 2011.

[20] Anpeng Wu, Kun Kuang, Ruoxuan Xiong, and Fei Wu. Instrumental variables in causal inference and machine learning: A survey. *ACM Computing Surveys*, 57(11):1–36, 2025.

[21] Fangzhao Wu, Ying Qiao, Jiun-Hung Chen, Chuhan Wu, Tao Qi, Jianxun Lian, Danyang Liu, Xing Xie, Jianfeng Gao, Winnie Wu, et al. Mind: A large-scale dataset for news recommendation. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 3597–3606, 2020.

[22] Tong Wang, K Sudhir, and Dat Hong. Using advanced llms to enhance smaller llms: An interpretable knowledge distillation approach. *arXiv preprint arXiv:2408.07238*, 2024.

[23] Kevin Lee. Generative brand choice. Technical report, Working Paper, 2024.

[24] Elea McDonnell Feit and Ron Berman. Test & roll: Profit-maximizing a/b tests. *Marketing Science*, 38(6):1038–1058, 2019.

[25] Alex P Miller and Kartik Hosanagar. An empirical meta-analysis of e-commerce a/b testing strategies. *The Wharton School, University of Pennsylvania*, 2020.

[26] Max H Farrell, Tengyuan Liang, and Sanjog Misra. Deep neural networks for estimation and inference. *Econometrica*, 89(1):181–213, 2021.

[27] Andrew Bennett, Nathan Kallus, and Tobias Schnabel. Deep generalized method of moments for instrumental variable analysis. *Advances in neural information processing systems*, 32, 2019.

[28] Amandeep Singh and Bolong Zheng. Causal regressions for unstructured data. In *Causal Representation Learning Workshop at NeurIPS 2023*, 2023. URL `https://openreview.net/forum?id=Zs3C7zytfp`.

[29] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

[30] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36, 2024.

[31] Rui Zheng, Shihan Dou, Songyang Gao, Yuan Hua, Wei Shen, Binghai Wang, Yan Liu, Senjie Jin, Qin Liu, Yuhao Zhou, et al. Secrets of rlhf in large language models part i: Ppo. *arXiv preprint arXiv:2307.04964*, 2023.

[32] Hamish Ivison, Yizhong Wang, Jiacheng Liu, Zeqiu Wu, Valentina Pyatkin, Nathan Lambert, Noah A Smith, Yejin Choi, and Hannaneh Hajishirzi. Unpacking dpo and ppo: Disentangling best practices for learning from preference feedback. *arXiv preprint arXiv:2406.09279*, 2024.

[33] Eirini Ntoutsi, Pavlos Fafalios, Ujwal Gadiraju, Vasileios Iosifidis, Wolfgang Nejdl, Maria-Esther Vidal, Salvatore Ruggieri, Franco Turini, Symeon Papadopoulos, Emmanouil Krasanakis, et al. Bias in data-driven artificial intelligence systems—an introductory survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 10(3):e1356, 2020.

[34] Jeremy Tien, Jerry Zhi-Yang He, Zackory Erickson, Anca D Dragan, and Daniel S Brown. Causal confusion and reward misidentification in preference-based reward learning. *arXiv preprint arXiv:2204.06601*, 2022.

[35] Carson Denison, Monte MacDiarmid, Fazl Barez, David Duvenaud, Shauna Kravec, Samuel Marks, Nicholas Schiefer, Ryan Soklaski, Alex Tamkin, Jared Kaplan, et al. Sycophancy to subterfuge: Investigating reward-tampering in large language models. *arXiv preprint arXiv:2406.10162*, 2024.

[36] Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

[37] Harrison Lee, Samrat Phatale, Hassan Mansoor, Kellie Ren Lu, Thomas Mesnard, Johan Ferret, Colton Bishop, Ethan Hall, Victor Carbune, and Abhinav Rastogi. RLAIF: Scaling reinforcement learning from human feedback with AI feedback, 2024. URL `https://openreview.net/forum?id=AAxIs3D2ZZ`.

[38] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[39] Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, pages 10835–10866. PMLR, 2023.

[40] Rafael Rafailov, Yaswanth Chittepu, Ryan Park, Harshit Sushil Sikchi, Joey Hejna, Brad Knox, Chelsea Finn, and Scott Niekum. Scaling laws for reward model overoptimization in direct alignment algorithms. *Advances in Neural Information Processing Systems*, 37:126207–126242, 2024.

[41] Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlíček, Agustín Piqueres Lajarín, Vaibhav Srivastav, Joshua Lochner, Caleb Fahlgren, Xuan-Son Nguyen, Clémentine Fourrier, Ben Burtenshaw, Hugo Larcher, Haojun Zhao, Cyril Zakka, Mathieu Morlon, Colin Raffel, Leandro von Werra, and Thomas Wolf. Smollm2: When smol goes big – data-centric training of a small language model, 2025. URL `https://arxiv.org/abs/2502.02737`.

[42] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1 (2):3, 2022.

[43] Lik Xun Yuan. distilbert-base-multilingual-cased-sentiments-student (revision 2e33845), 2023. URL `https://huggingface.co/lxyuan/distilbert-base-multilingual-cased-sentiments-student`.

[44] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *ACM computing surveys (CSUR)*, 54(6):1–35, 2021.

[45] Hui Zou and Hao Helen Zhang. On the adaptive elastic-net with a diverging number of parameters. *Annals of statistics*, 37(4):1733, 2009.

[46] Jianqing Fan and Yuan Liao. Endogeneity in high dimensions. *Annals of statistics*, 42(3):872, 2014.

A Related work

Our research intersects three core areas: (1) the use of language models in science and business applications, (2) econometric approaches to causal inference in machine learning, and (3) techniques and challenges for aligning large language models (LLMs). We summarize key contributions in each domain and highlight how our work extends current boundaries, particularly in aligning LLMs using observational data subject to confounding.

A.1 LLMs in science and business applications

Recent work underscores the value of LLMs across the sciences—from electronic health records Wu et al. [7] to astronomical data Wang et al. [8] and social-science corpora—yet it rarely addresses causal identification or the pitfalls of learning from historical data. In business settings, most studies pair LLMs with randomized feedback rather than historical data: Ye et al. [4] combine LLMs with adaptive experimentation (UCB) to optimize headline click-through rate (CTR), and Angelopoulos et al. [9] fine-tune on A/B outcomes to generate subject lines that beat human baselines. These results showcase the power of experimental supervision while leaving the role of observational data underexplored.

Other lines of research focus on knowledge transfer and demand prediction. For example, Wang et al. [22] propose a distillation framework in which smaller models learn response behaviors from larger LLMs, assuming that the teacher model's outputs serve as a reliable proxy for optimal performance. In a different application area, Lee [23] addresses demand prediction for new products. The proposed method first estimates consumer preferences using a structural model and then uses an LLM to map textual product descriptions to these estimated preferences.

Together, these works highlight the promise of LLMs in business settings. However, aside from the last study, which focuses on demand prediction rather than content generation, these methods primarily rely on supervision signals obtained from controlled experiments, which are costly and limited in scope [24, 25], or on synthetic feedback that may reflect the biases of the teacher model. In contrast, our work investigates how to fine-tune LLMs using abundant observational data, while explicitly addressing the confounding factors that can mislead model learning.

A.2 Causal inference with machine learning and econometrics

The intersection of causal inference and machine learning has received significant attention in econometrics. Chernozhukov et al. [16] introduce Double Machine Learning (DML), using orthogonalized moment conditions and cross-fitting to reduce bias from regularization in high-dimensional settings. Farrell et al. [26] extend this framework with theoretical analysis for deep neural networks in semiparametric models. However, both methods assume exogeneity, limiting their utility when confounding is endogenous. To address this, researchers have proposed adversarial and instrumental variable (IV) methods. Dikkala et al. [17] and Bennett et al. [27] apply adversarial GMMs for nonparametric IV regression, framing estimation as a minimax game over moment conditions. More recently, Singh and Zheng [28] develop a flexible IV estimator to estimate policy effects for unstructured data, such as text and images.

Our work contributes to this line of research by adapting IV-based ideas for generative modeling with LLMs. Instead of estimating structural parameters, our goal is to deconfound reward signals used in LLM fine-tuning, ensuring the model aligns with causal rather than spurious objectives.

A.3 LLM alignment and reward modeling

A core challenge in LLM development is aligning models with human intent and utility. Reinforcement learning from human feedback (RLHF) has emerged as a standard approach [29], with techniques like PPO and DPO fine-tuning models based on preference data [30, 31]. Recent empirical studies [32] reveal that PPO often outperforms other methods in capturing nuanced user preferences, particularly when high-quality reward models are used. Yet, reward modeling remains susceptible to biases. As pointed out by Ntoutsi et al. [33], human-labeled data itself could suffer from bias. Furthermore, LLMs often overfit to artifacts like response length or stylistic sycophancy [34, 35]. To address these issues, several works have developed causal reward modeling frameworks. However,

these methods typically assume counterfactual invariance, that the reward should not change when confounding attributes are perturbed—an assumption that often fails in practice. ODIN [10] removes known confounders (e.g., length) from the learned reward. Wang et al. [11] train rewards to satisfy counterfactual invariance by construction. More recently, Srivastava et al. [12] propose robust reward modeling method that enforces counterfactual invariance via counterfactual data augmentation. While the counterfactual invariance assumption may hold for LLM-generated answers evaluated by humans, it does not hold in many real-world applications, which we discuss more below.

Formally, these approaches assume that the true reward function satisfies

$$f(\boldsymbol{F}, \boldsymbol{C}) = f(\boldsymbol{F}),$$

where $\boldsymbol{F}$ represents meaningful input features and $\boldsymbol{C}$ denotes confounding variables. This assumption enforces that rewards are entirely invariant to changes in $\boldsymbol{C}$. While effective in controlled alignment scenarios, this assumption is overly restrictive for business applications, where confounders like price, seasonal effects, or audience composition may influence both input features and outcomes. In these cases, confounders should not be ignored outright but rather correctly accounted for. Our method, DECONFOUNDLM, adopts a more flexible formulation. Instead of assuming that $\boldsymbol{C}$ has no effect on the reward, we aim to estimate and remove the spurious influence of $\boldsymbol{C}$ on observed outcomes. This approach enables the model to align with causal drivers of business performance rather than with misleading correlations in the data.

These contributions collectively underscore the need for more causally grounded, scalable alignment methods, especially when experimental data is scarce. By integrating econometric techniques with LLM reward modeling, our work provides a principled framework for aligning models trained on observational logs.

B Problem setup and background

Fine-tuning LLMs to align their outputs with user preferences is a common approach for enhancing their performance. This process typically relies on *labeled preference data*, which may be collected through human annotations [36], automated feedback mechanisms (e.g., RLAIF [37]), or structured reasoning tasks (e.g., Guo et al. [38]). Two major paradigms are commonly used to incorporate preference data into LLM training: (i) Reward Modeling followed by Reinforcement Learning, and (ii) Direct Preference Optimization.

In the former, a reward model $r_\phi(x, y)$ is first trained to predict human preferences between outputs given an input x. The model is typically trained using pairwise comparisons, optimizing a Bradley-Terry likelihood:

$$\mathcal{L}_{\text{RM}}(\phi) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}} \left[\log \sigma \left(r_\phi(x, y_w) - r_\phi(x, y_l)\right)\right],$$

where σ is the logistic sigmoid. Once trained, this reward model is used to fine-tune the language model $\pi_\theta(y \mid x)$ using reinforcement learning algorithms such as Proximal Policy Optimization (PPO), which maximize expected reward while regularizing against a reference policy:

$$\max_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D},\, y\sim\pi_\theta(\cdot|x)} \left[r_\phi(x, y) - \beta\, \mathrm{KL}\left(\pi_\theta(\cdot \mid x) \,\|\, \pi_{\text{ref}}(\cdot \mid x)\right)\right].$$

In contrast, *Direct Preference Optimization (DPO)* bypasses reward modeling entirely and directly updates the policy to prefer higher-rated responses using a contrastive objective over preference pairs [30].

In industrial applications such as optimizing click-through rates for headlines, boosting booking rates on rental platforms, or improving adherence in health messaging, the gold standard for evaluating outcomes is randomized controlled trials, which allow unbiased estimation of causal effects. However, such experiments are expensive and often infeasible in practice. Meanwhile, organizations often have abundant *observational data*: historical logs of content such as page titles, messages, or headlines and their associated outcomes. This data can be used directly for fine-tuning, either by training a reward model or by constructing preference pairs for methods like DPO. The challenge, however, is that observational data is subject to confounding: unobserved variables may influence both the textual content and the observed outcome, leading to spurious correlations. For example, consider an LLM deployed at Airbnb to generate listing titles aimed at increasing reservation rates. Historical

data may show that listings with the word *"affordable"* in the title perform better. However, this could reflect the underlying confounding effect of the price, as lower-priced listings generally get higher reservations. A model fine-tuned naively on this data may learn to associate *"affordable"* with success in all contexts, leading to unsuitable generations. Say you are asking the model to generate a title for a luxury riverside property, and the model generates `''Affordable log chalet - perfect for solo travelers''`!

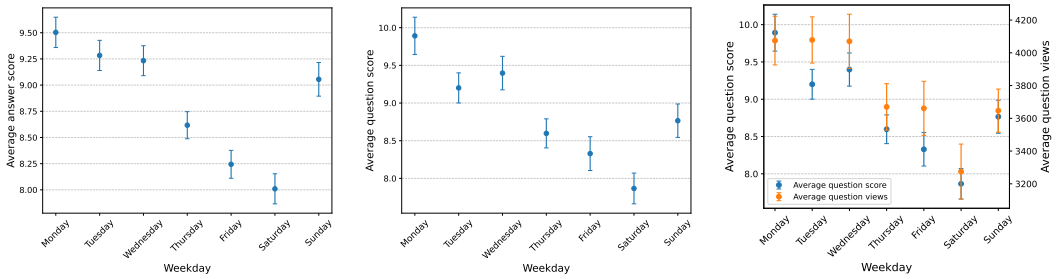
This is a clear instance of reward misspecification: the model learns to optimize a proxy signal that only partially reflects the true objective. While prior work [39, 40] has investigated reward over-optimization in both classical RLHF and Direct Alignment settings, these studies have largely focused on empirical scaling behavior and optimization dynamics. In this work, we take a step further by analyzing the role of confounding and causal misalignment in fine-tuning large language models using observational data. In the following section, we investigate how relying on historical preference data can lead to biased and potentially flawed model fine-tuning, by examining a real-world case study.

C The Monday experiment: An example of confounding in observational data

In this section, we present an example illustrating how confounding can lead to reward misspecification when fine-tuning language models using historical data. Specifically, we construct a dataset similar to that used by Askell et al. [13], based on data from the Academia Stack Exchange. In their study, the authors fine-tune a question-answering model and, in one step of their fine-tuning, Preference Model Pre-Training (PMP), use historical data to guide learning. They treat answer scores as preference signals and train the model to prefer higher-scored answers in cases where multiple answers are available for a question. This PMP step is followed by fine-tuning with human feedback, to ensure the alignment of the model's preferences with human judgments. In our experiment, we investigate what happens when human feedback is unavailable and only observational data is used for fine-tuning.

While using the scores can signal which answer is more helpful, these scores are not the outcomes of randomized experiments, rather could be affected by user engagement patterns. For example, answers posted earlier may receive more views and thus more votes. One confounder we investigate is periodicity in platform engagement across different weekdays. To investigate this, we analyze the average answer scores by weekday. As shown in Figure 2a, answers posted on Mondays receive significantly higher scores than those posted on Fridays. While one might speculate that this might be causal and reflect differences in writing quality, we observe a similar pattern in the average scores of questions themselves (Figure 2b), suggesting that broader engagement trends may be at play.

We further examine the number of views per question as a proxy for user exposure. Since the dataset does not include view counts for individual answers, we cannot directly assess the effect of exposure at the answer level. Figure 2c displays both question views and scores over time. The strong correlation between the two suggests that the observed temporal trends are more likely driven by fluctuations in user activity than by differences in content quality.



(a) Average answer score by weekday

(b) Average question score by weekday

(c) Average question score and views by weekday

Figure 2: Weekly temporal patterns in Stack Exchange engagement. User scores and views exhibit strong weekday effects, with higher engagement early in the week.

To test whether this bias can influence model behavior, we simulate a fine-tuning setup similar to that of Askell et al. [13]. We construct answer pairs based on user scores and designate the higher-scoring answer as preferred. For answers posted on Mondays, we prepend a neutral `"Happy Monday!"` phrase to introduce a content marker correlated with engagement rather than quality. We then evaluate model generations on 3000 held-out questions and count how often the words "Happy" and "Monday" appear. Table 1 summarizes the results. The base pre-trained model rarely generates these terms. The supervised fine-tuned (SFT) should ideally capture the distribution in the data (generation temperature is set to 1). Our results show a frequency of 13.7% ± 0.2% for the first word "Happy" which is consistent with the distribution in the data ($\sim 1/7$). In comparison, the DPO model generates "Happy" in 21.2% ± 0.8% and "Monday" in 11.8% ± 0.8%, representing substantial increases of approximately 7.5 and 2.7 percentage points, respectively. To assess whether these increases are statistically significant, we perform independent two-sample t-tests over the 25 generation rates from each model. The difference in "Happy" usage is highly significant ($p = 6.5 \times 10^{-10}$), and the increase in "Monday" usage is also statistically significant ($p = 3.4 \times 10^{-3}$). These results show that the model has internalized and amplified a spurious temporal signal. Additional details of this experiment, as well as further details about data and training characteristics, are provided in Appendix C.

Table 1: Mean percentage (standard error) of generations containing "Happy" and "Monday" across 5 generation seeds for the base model, and 25 runs (5 fine-tuning seeds × 5 generation seeds) of SFT and DPO fine-tuning. DPO fine-tuning significantly amplifies the spurious weekday signal.

Model	**Generations per Run**	**Num. Runs**	**With "Happy" (%)**	**With "Monday" (%)**
Base Model	3000	5	1.13 (0.04)	0.07 (0.02)
SFT Model	3000	25	13.69 (0.21)	9.05 (0.19)
DPO Model	3000	25	**21.22 (0.77)**	**11.78 (0.81)**

This case highlights how confounding variables in observational datasets can lead to reward misspecification and unintended behavior in fine-tuned models. Without accounting for causal structure, models may learn to exploit spurious signals that correlate with success, even when they do not contribute to genuine task quality.

Data. To replicate and extend the setup of Askell et al. [13] we use data from the Academia Stack Exchange. The dataset contains 104,426 question-answer pairs. We retain only those questions with multiple answers, reducing the data to 82,737 answer instances. To reduce memory usage during training, we further restrict to questions and answers with fewer than 180 words, yielding 33,194 answers across 14,319 questions. Of these, 4,937 answers were written on a Monday.

We split the questions into three groups: 5,000 for supervised fine-tuning (SFT), 3,000 for testing, and the remainder for reward-based fine-tuning. For each question in the fine-tuning subset, we form ordered answer pairs by comparing scores and labeling the higher-scored answer as preferred. We cap the number of pairs per question at 10 to prevent imbalance. This yields 11,886 pairs, where we find a notable weekday skew: in pairs with only one Monday answer, 958 have the Monday answer as preferred, while 887 have it as rejected, hinting at temporal confounding.

Fine-tuning setup. We use the 360M parameter `SmolLM2-Instruct` [41] model as the base and perform two-stage fine-tuning.

Supervised Fine-Tuning (SFT). The SFT step uses the answer text as the assistant response and the corresponding question as input. Training is done for 1 epoch with a batch size of 8 and a learning rate of 2×10^{-4} using the AdamW optimizer (8-bit). We apply LoRA [42] with rank 16 and dropout 0.1. Inputs are tokenized using a custom prompt template with a 512-token sequence limit.

Direct Preference Optimization (DPO). The DPO stage initializes from the SFT checkpoint and fine-tunes using the constructed answer preference pairs. We use a β of 0.1 and train for up to 4 epochs with a batch size of 8. LoRA is applied with rank 8. The maximum prompt and completion lengths are 256 and 512 tokens, respectively.

Generations for evaluation are performed on a held-out set of 3,000 questions, and model outputs are assessed for lexical artifacts. For the base model, which is fixed and not subject to any fine-tuning

variability, we introduce randomness only through the generation process by using 5 different random seeds. In contrast, both the SFT and DPO models are subject to randomness in fine-tuning as well as generation. Specifically, we fine-tune each model using 5 different random seeds for initializing the model head, and then generate outputs from each trained model using 5 different generation seeds. This results in 25 runs per model, each producing 3000 generations.

To qualitatively assess the effect of fine-tuning, Table 2 presents representative generations from the base, SFT, and DPO models for a single held-out question. This example illustrates how fine-tuning shapes model behavior and provides evidence that the fine-tuning process is likely implemented correctly. The base model produces a generic answer formatted in Markdown (e.g., using ** for bold), reflecting conventions likely present in its pretraining data. In contrast, the SFT model aligns more closely with the domain-specific formatting patterns found in the training set, namely, HTML tags. It adopts a tone more typical of answers on the target forum. The DPO model builds upon this, showing further alignment with patterns found in high-scoring answers: it employs richer formatting (e.g., `<strong>`) and incorporates external references, likely reflecting learned associations between such features and answer quality.

Compute. All experiments for this setup were conducted on an NVIDIA RTX A6000 GPU. The Supervised Fine-Tuning (SFT) stage is relatively lightweight and completes in approximately 10 minutes. In contrast, the Direct Preference Optimization (DPO) stage is more computationally intensive due to its iterative training on preference pairs and takes around 1 hour to run per seed.

D Upworthy experiment details

We follow a similar data processing approach to that of Ye et al. [4], using the Upworthy dataset. The full dataset includes 150,817 headline-image "packages" across 32,487 A/B tests. Since some tests involve variation in both headlines and images, we restrict our analysis to headline-only tests where the image remains fixed. This filtering yields 17,682 headline-only tests comprising 77,245 packages.

To construct the experimental dataset, we generate all possible headline pairs within each A/B test and retain only those with a statistically significant difference in click-through rate (CTR) at the 5% level. This results in 41,624 headline pairs covering 27,745 packages. We split these into training (60%), validation (20%), and test (20%) sets, while ensuring no headline appears in more than one split to avoid data leakage. The final dataset includes 24,842 training pairs, 8,395 validation pairs, and 8,387 test pairs.

These statistically significant pairs form the basis of our experimental setting. To simulate a non-experimental setting, we derive a corresponding observational dataset. For each headline test in the training set, we randomly retain only one package and discard the counterfactual. This results in 8,499 training packages, representing approximately 26% of the total packages. This setup reflects a typical historical logging scenario, where only observed outcomes are available. Table 3 provides summary statistics of the experimental and observational datasets.

Before moving on to the modeling details, we briefly highlight a potential confounder that can affect observational CTRs: temporal variation in user engagement and topic salience. The probability that a user clicks on a given package depends not only on the quality or attractiveness of the headline, but also on who the viewers are and how relevant or important the topic is at the time. For instance, if a major sporting event occurs, the site may receive a surge of sports fans, whose preferences disproportionately influence overall CTRs. Similarly, politically themed headlines may receive more engagement during election periods. Figure 3a shows the average CTR by month for three data subsets: experimental training packages, observational training packages, and observational validation packages. We observe a clear temporal pattern, with certain months getting substantially higher CTRs. Moreover, the CTR trends are highly correlated across subsets (pairwise correlations of monthly averages are between 96% and 97%), suggesting that these fluctuations are systematic rather than random.

This raises a concern for models trained directly on raw CTRs. They may overfit to superficial, time-related artifacts rather than learning meaningful properties of headline quality. As previously discussed, variations in CTR may partly reflect changes in the user population and taste rather than differences in content effectiveness. Figure 3b provides evidence of these changes, showing substantial variation in the number of impressions per package across months. Notably, there is a

Table 2: Comparison of model responses for a given question.

Question	Model	Generated Answer
" I've been a Computer Sciences student for 4 years now, give or take, and my experience with emailing teachers/professors (teachers from now on) has been either of two cases: - they reply to my entire email - they read the last sentence or paragraph and reply to that. Generally, if a teacher fits in one of those categories, they always respond like that. However, for me it's quite frustrating to receive an email back (within 5 minutes, an hour, a day) with a reply, to only one point, and then having to mail back with my other questions, removing one question at a time. To note: my e-mails are always very respectful, neatly written, and without spelling or other mistakes. What can I do to get a higher percentage of questions-answered-per-email response? "	**Base Model**	1. **Include specifics about the difficulties you face**: Instead of a general statement, explain the difficulties youŕe facing. This will show that youŕe willing to help by providing more details. 2. **Offer solutions to the problem**: Provide some strategies or tips to help teachers improve their email responses to your questions. This will show that youŕe proactive and interested in finding a solution to the problem. 3. **Provide clear examples of your goals and objectives**: Be specific about what youŕe trying to achieve by improving the response rate. For instance, you could say that youd́ like to see more direct responses or more personalized responses from the teachers. 4. **Highlight any potential benefits to the school**: While you mentioned that the teacher-student ratio might be a problem, you could also mention that the improvement in response rate could positively affect the students, such as by having more individualized support. 5. **Emphasize the time and effort required**: Acknowledge that sending multiple emails back and forth might be time-consuming, but also express that the benefits of getting more responses are worth the extra effort. Example of how you could write a specific paragraph to demonstrate this: "Iḿ frustrated that Iv́e received responses to my
	SFT Model	Happy Monday! If my general attitude is that you should be able to reply to a very large number of emails with a single email, your teacher should answer the first few emails, since they don't need to spend a lot of time answering those. On the other hand, if your teacher is very particular about your answers, you might want to offer more details about what you're trying to accomplish in your course, and what kind of questions you have and are trying to get answered.
	RL (DPO)	Happy Monday! The first point is that people generally answer more questions than they reply to in the course of an email. In a rough order, the answers are: - Less frequently, but they read the last sentence or paragraph and then reply to that: Reciprocity with you. It's nice to hear from you again and it's worth the extra effort for them to do so. - Less frequently, but you read the last sentence or paragraph and then reply to that: Reciprocity. Ask a question and get answers. - Less frequently, but you don't read the last sentence or paragraph and reply to that: ...

marked increase in impressions toward the end of 2023, coinciding with the U.S. election period. These fluctuations suggest that the volume and potentially the composition of website traffic change over time. As a result, shifts in user demographics or interests could introduce biases into the observed CTRs, potentially misleading models trained on such observational data.

(a) Monthly average CTRs across different data splits.

(b) Monthly average impressions per package.

Figure 3: Temporal patterns in user engagement. The left plot shows variation in average click-through rates (CTR) across months, while the right plot shows the number of impressions per package, indicating changes in user traffic volume.

Reward modeling. To train reward models, we use a prompting structure where the model is asked to generate a headline for a given news abstract:

```
System: You are an editor of a news website.
Your task is to generate a headline for each news article that
will attract the most readers. The headline should be less than 40 words.
Only respond with the headline.
User: The news abstract is ‘{lede}‘ News posted at {created_at}
Assistant: {headline}
```

We use models from the Pythia suite [15] to generate embeddings, specifically extracting the representation of the final token in each output. A classification or regression head is added on top of this embedding to predict outcomes (CTR or preference), and an L_2 regularization parameter λ is tuned to manage overfitting, as detailed in the main text.

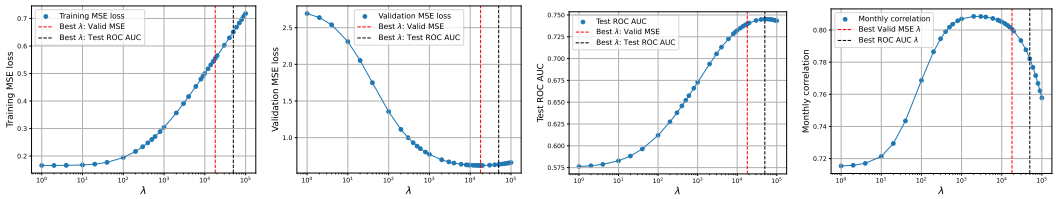
Table 3: Summary statistics of the Upworthy dataset for experimental and observational settings.

Statistic	**Upworthy Data**	
Total headline-only A/B tests	17,682	
Total packages	77,245	
	Experimental Data	**Observational Data**
Statistically significant pairs	41,624	–
Packages in significant pairs	46,330	–
Training pairs	24,842	–
Training packages	27,745	7,285
Validation pairs	8,395	–
Validation packages	7,527	2,079
Test pairs	8,387	8,387

Importance of regularization. We study the role of regularization in observational learning and find that strong regularization is critical for generalization. As shown in Figures 4b and 4c, optimal validation loss occurs at $\lambda = 18{,}000$, yet the best test ROC AUC is achieved at $\lambda = 50{,}000$. This discrepancy suggests that in the presence of confounding factors, tuning hyperparameters solely based on validation loss may not suffice. The model may overfit to patterns influenced by spurious

correlations in the validation data, rather than learning features that generalize causally to unseen headline comparisons. Figure 5a shows that this gap holds across model sizes: stronger regularization consistently yields better test performance than what validation loss would suggest. We further find that larger models generally require stronger regularization for optimal test performance. This observation implies that using a fixed regularization setting across models of different sizes is suboptimal. Figure 5b demonstrates this by plotting test performance against model size under fixed regularization levels. The figure shows a non-monotonic effect, larger models begin to overfit more if regularization is kept constant. These results emphasize the need to scale regularization appropriately with model capacity in order to maintain generalization, which is often overlooked in practice.

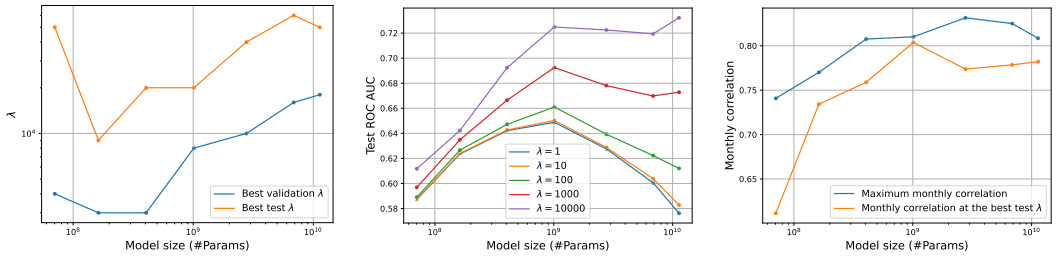
While these results underscore the critical role of regularization, they also raise a practical challenge when access to held-out experimental data for tuning hyperparameters is often limited. In such cases, alternative strategies are needed to remove the effect of confounders. We address this issue in Section 3, where we introduce a method for explicitly correcting for confounding effects in observational fine-tuning.



(a) Training MSE vs. λ (b) Validation MSE vs. λ (c) Test ROC AUC vs. λ (d) Monthly corr. vs. λ

Figure 4: Effect of regularization strength (λ) on different evaluation metrics for the Pythia-12B model.

Temporal pattern overfitting. As discussed before, temporal variation in CTRs is a potential confounder in observational data. To assess how much models internalize these patterns, we compute the correlation between monthly average CTR estimates on the validation set and observed monthly CTRs in the training data. Figure 4d shows this correlation across values of λ for the Pythia-12B model. We observe that moderate regularization improves alignment with temporal patterns, but higher regularization reduces it. Interestingly, the λ that yields the best test performance comes well after this drop, indicating that suppressing temporal patterns helps the model on the causal evaluation of headlines. This trend holds across model sizes. As shown in Figure 5c, models consistently show lower temporal correlation at their optimal test-time λ, further suggesting that failing to effectively account for confounding patterns can impair generalization performance.



(a) Optimal λ by model size (validation vs. test) (b) Test ROC AUC with fixed regularization (c) Monthly correlation

Figure 5: Regularization dynamics across model sizes. (a) Models require larger regularization to achieve optimal test performance. (b) Using a fixed regularization setting leads to non-monotonic scaling performance. (c) Models show lower correlation with temporal engagement patterns at their optimal test-time λ.

Compute. The most computationally intensive part of this experiment is generating embeddings using models from the Pythia suite. We extract the final-token representations, which serve as inputs to

the reward models. These embedding computations are performed on an AMD Radeon 7900 GPU. For the largest model used in our experiments, Pythia-12B, the embedding generation takes approximately 12 minutes for the observational dataset and about 1.5 hours for the experimental dataset, which is larger. Once embeddings are obtained, training the reward models with a classification or regression head is relatively lightweight and runs efficiently on the Intel(R) Xeon(R) Gold CPU @ 2.90GH.

E Details of simulation experiments

For our synthetic experiments, we use the MIND (Microsoft News Dataset) [21], which contains 160,000 English news articles, each with a headline and article body. To simulate user engagement, we construct synthetic performance scores (interpretable as click-through rates) for the article headlines using equations (6), (7), and (8). To find the sentiment of each headline in the data, we use the sentiment analysis model from [43].

To ensure domain consistency, we focus on the sports category, which includes 54,553 articles, the largest among all categories. The data is split as follows: 20,000 articles for Supervised Fine-Tuning (SFT), 10,000 for Reward Modeling (RM), 3,000 for reward validation, 10,000 for Proximal Policy Optimization (PPO), and the rest for testing.

Example 1 (Partially Linear Regression)**.** Consider a partially linear model where the confounder p_i, e.g., the price in Airbnb listings, enters linearly (Similar to the examples from Chernozhukov et al. [16]):

$$y_i = g(T_i, \tilde{\boldsymbol{F}}_i) + \alpha p_i + \epsilon_i, \tag{5}$$

Here, $p_i \in \boldsymbol{C}_i$ is an observed confounder. In the Airbnb example, when optimizing titles to improve reservation rates, price may strongly affect y_i and also correlate with certain title patterns (e.g., "affordable"). Estimating g accurately thus requires adjusting for p_i to avoid spurious correlations.

Confounding In this section, we describe how we model the confounding. Across both scenarios, we model the outcome as:

$$y_i = s(T_i) + 0.1\, p_i + \nu_i, \tag{6}$$

where p_i is the confounder, and $\nu_i \sim \mathcal{N}(0, 0.1)$ represents observational noise.

We vary how p_i is constructed across two settings:

- *Orthogonal confounding.* The confounder p_i is independent of the sentiment $s(T_i)$, making its effect easier to isolate and remove. Specifically:

  $$p_i = \mathbb{1}(\text{title mentions West Coast team}) + 2 \cdot \mathbb{1}(\text{Central team}) + 3 \cdot \mathbb{1}(\text{East Coast team}) + \epsilon_i, \tag{7}$$

  where $\epsilon_i \sim \mathcal{N}(0, 0.5)$. This reflects a hypothetical bias where East Coast teams are generally more popular and draw higher engagement regardless of the title's quality.
- *Entangled confounding.* Here, popularity p_i is correlated with the sentiment of the news abstract, mimicking a setting where emotional salience of the topic and engagement co-vary. For instance, sad events may draw more audience to the platform and lead to increased engagement. We model this with:

  $$\begin{aligned} p_i = {} & \mathbb{1}(\text{title mentions West Coast team}) + 2 \cdot \mathbb{1}(\text{Central team}) + 3 \cdot \mathbb{1}(\text{East Coast team}) \\ & - 10.5 \cdot s(\text{abstract}) + \epsilon_i. \end{aligned} \tag{8}$$

These synthetic scenarios allow us to explicitly test whether models can recover the true effect of sentiment when the observed performance signal is partially corrupted by a structured confounder.

Supervised Fine-Tuning. We fine-tune a language model using SFT, where the model is prompted to generate engaging headlines from article abstracts. The prompting structure is:

```
System: You are an editor of a news website. Your task is to
generate a headline for each news article that will attract the most
readers. The headline should be less than 30 words. Only respond with
the headline.
User: The news abstract is ‘{abstract}‘
Assistant: {headline}
```

The base model is `HuggingFaceTB/SmolLM2-360M-Instruct`, fine-tuned with LoRA (rank 16, α=32, dropout=0.1) for one epoch. We use a learning rate of $2e{-}4$ and batch size of 8.

Reward modeling. To train reward models on the synthetic performance scores, we perform hyperparameter tuning over several learning rates: {2e-4, 6e-4, 8e-4, 1e-3, 2e-3}. Based on prior findings [29], we limit training to one epoch to avoid overfitting.

Generation results. Table 4 summarizes the average sentiment of generated headlines and the frequency of team name mentions, which serve as a proxy for reliance on the popularity-based confounder. We discussed these results in Section 3.1.

Table 4: Comparison of models under two confounding scenarios. The table reports the mean sentiments and the number of generated titles mentioning teams by region. Models are tested on 3,000 headline generations. Note that the reported results for the first four models are identical across both scenarios, as they do not rely on the observed performances; the difference between the two scenarios lies solely in how the observed performance is constructed.

Model	**Scenario 1: Orthogonal**				**Scenario 2: Entangled**			
	Sent.	W	C	E	Sent.	W	C	E
Base Pre-Trained Model	0.684	177	795	717	0.450	177	795	717
SFT Model	0.716	159	634	610	0.716	159	634	610
Model with only sentiment	0.950	165	714	728	0.960	187	697	720
Model with sentiment + noise	0.934	172	699	664	0.958	174	716	699
RL w/ observed performance	0.956	214	914	1016	0.735	186	908	1041
RL w/ pop. in text	0.932	158	655	638	0.718	166	676	629
RL w/ pop. in layer	0.935	178	679	661	0.800	189	829	835
ODIN [10]	0.934	178	729	680	0.807	249	1018	1063
DECONFOUNDLM-IV	0.939	181	692	682	0.937	170	736	694

Reward-sentiment correlation. While the main text highlights how different models affect the sentiment and team references in generated headlines (Table 4), here we focus on how well the trained reward models track sentiment directly. Table 5 shows the average Pearson correlation between predicted rewards and sentiment scores in the reward validation set, under two confounding scenarios. In the orthogonal case, observed performance is positively correlated with sentiment, allowing most models to achieve a positive correlation between their reward estimates and sentiment. However, in the entangled case, where the confounder (e.g., team popularity) effect is entangled with the outcome, this relationship breaks down. Most of the models that do not account for the confounder, or attempt to include it through text features or final-layer embeddings, fail to maintain a positive correlation between predicted rewards and sentiment. In contrast, DECONFOUNDLM-IV remains robust across both scenarios, maintaining a strong positive correlation.

Compute. Our simulation experiments were run using two types of GPUs: NVIDIA RTX A6000 and AMD Radeon 7900. For each combination of training seed and learning rate, reward modeling takes approximately 3–5 minutes on either GPU. However, the PPO fine-tuning stage is significantly more time-consuming, requiring about 2–3 hours to complete per setting.

Table 5: Correlation between predicted rewards and sentiment across two confounding scenarios. Each cell shows the Pearson correlation on the train and validation sets. The reported results for the first two models are identical across both scenarios, as they do not rely on the observed performances.

Model	**Scenario 1: Orthogonal**		**Scenario 2: Entangled**	
	Train	Valid	Train	Valid
Model with only sentiment	0.913	0.872	0.913	0.872
Model with sentiment + noise	0.910	0.891	0.807	0.802
RL w/ observed performance	0.880	0.859	-0.073	-0.079
RL w/ pop. in text	0.839	0.827	-0.261	-0.253
RL w/ pop. in layer	0.862	0.841	0.623	0.627
ODIN [10]	0.654	0.641	-0.536	-0.542
DECONFOUNDLM-IV	**0.915**	**0.886**	**0.898**	**0.867**

F Impacts and assumptions of our framework

Our framework enables the use of observational data to align large language models (LLMs), thereby opening new possibilities for alignment with significant potential for positive social impact. As discussed in the main body of the paper, there are many real-world scenarios where conducting randomized experiments on content and messaging is infeasible, while firms often possess extensive historical observational data. In such cases, leveraging this data can substantially improve the alignment of LLMs with organizational or societal objectives. Consider, for example, a messaging system designed to improve medication adherence among patients. While running an experiment might be challenging due to engineering and ethical challenges, optimizing such a system using observational data could lead to substantial improvements in health outcomes. However, as with any machine learning paradigm that seeks to optimize a performance metric, this approach also presents challenges. As highlighted in prior work [44], various forms of bias can influence the outputs of machine learning models.

Our framework specifically targets biases arising from confounders that influence both the treatment and the outcome. While we have not yet conducted empirical evaluations of the bias correction component with respect to mitigating group-level disparities, the proposed method can be used to account for societal factors that might otherwise lead a model to prefer one textual input over another based on irrelevant or unfair criteria. Furthermore, researchers and practitioners must consider heterogeneity in individual responses to different texts to prevent the model from unintentionally encoding or amplifying structural disparities. For example, in a mobile health messaging application, if a particular message yields high adherence overall but performs poorly for a specific subgroup, it is crucial to incorporate recipient characteristics into the model to ensure equitable outcomes and avoid disproportionately favoring majority groups.

Turning to the theoretical underpinnings of our framework, prior work (see Section A.3) often assumes that confounders have no effect on the outcome, implying a functional form $f(\boldsymbol{F}, \boldsymbol{C}) = f(\boldsymbol{F})$. However, this assumption may not hold in practice, especially in business settings where variables such as price are important drivers of outcomes. In contrast, our approach allows for a more realistic representation of the data-generating process, formulated as follows:

$$\begin{aligned} y_i &= f(\boldsymbol{X}_i) + \epsilon_i \\ &= f(T_i, \tilde{\boldsymbol{X}}_i) + \epsilon_i \\ &= g(T_i, \tilde{\boldsymbol{F}}_i) + h(\boldsymbol{C}_i) + \epsilon_i. \end{aligned} \tag{9}$$

This formulation allows confounders to have a meaningful effect on outcomes, rather than assuming that outcomes are independent of confounder values. To ensure tractability, we impose two assumptions within our framework. First, we assume exogeneity of the error term conditional on the observed covariates, that is, $\mathbb{E}[\epsilon_i | \boldsymbol{X}_i] = 0$. This assumption is commonly made in empirical research involving high-dimensional covariates [45], though it is not without limitations. As discussed in [46], even in high-dimensional settings, incidental or unintentional endogeneity can arise due to selection bias or model misspecification. Second, we assume a separable functional form in the final line

of Equation 9, in which the effects of the confounders and the remaining variables are additively decomposed. Importantly, the model remains flexible enough to capture interactions between g and h through their shared inputs, as illustrated in the entangled case described in Section 3.1.

While our framework introduces greater flexibility than prior approaches, we acknowledge the limitations of these assumptions. The authors are currently working on developing a more general framework that further relaxes these conditions.