Advancing Generative AI via the Science of Data

by

Yuzheng Hu

Dissertation

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois Urbana-Champaign, 2026

Urbana, Illinois

Doctoral Committee:

Assistant Professor Han Zhao, Chair
Assistant Professor Jiaqi Ma
Professor David Forsyth
Professor Tong Zhang
Chiyuan Zhang, Google
Professor Sewoong Oh, University of Washington

Abstract

Data has become a central determinant of progress in generative AI: it drives model capability, shapes model behavior, and increasingly defines the limits of what can be achieved. We argue that advancing generative AI therefore requires treating data not as a passive training input, but as a first-class object of study. To this end, we develop theory and methods along two complementary directions: data attribution, which asks how data influences model behavior, and data privacy, which asks how sensitive data can be utilized under formal privacy constraints.

On the attribution side, we first develop a unified theory of random projection for influence functions, establishing when and why projection preserves influence in unregularized, regularized, and Kronecker-factored settings. This provides a theoretical foundation for scaling influence functions to large-scale generative models. We then extend attribution to online reinforcement learning (RL) by proposing a local framework for PPO that attributes checkpoint updates to the recent rollout buffer. This framework provides new tools for understanding learning dynamics and improving the training efficiency of online RL.

On the privacy side, we introduce the notion of empirical privacy variance, showing that the same formal differential privacy (DP) guarantee can correspond to markedly different empirical privacy behavior in practice. We further quantify how hyperparameters shape empirical privacy and develop privacy-aware hyperparameter tuning strategies in DP training. We then propose a hierarchical framework for differentially private synthetic text generation, augmented with a post-training recipe, anchored reinforcement learning, to improve the accuracy of conditional generation.

Overall, this thesis advances generative AI through the science of data: by quantifying the value of data, optimizing its use, preserving privacy, and unlocking the value of sensitive data through synthetic data generation.

To my family, for their love and support.

Acknowledgements

The journey began at a time of deep uncertainty.

After two discouraging research experiences in machine learning theory, I began to question the meaning of pursuing a PhD. At one point, I seriously considered abandoning the application process altogether. In September 2020, during that moment of hesitation, I had a life-changing conversation with Edward Suning Tian. That conversation reshaped my understanding of what applying for a PhD in the United States could mean. It was not merely a pursuit of academic credentials, but an entry point into a much broader world.

Edward reminded me that AI represents a once-in-a-generation opportunity, and that two unsuccessful attempts in a small subfield of it were insignificant compared to the broader landscape ahead. More importantly, he recognized in me qualities he valued: intellectual independence, curiosity, and a tendency to question prescribed paths. Through his generosity, I was introduced to James Ding, and together they enthusiastically connected me with two senior figures in AI, Fei-Fei Li and Dawn Song. Having the opportunity to speak with these inspiring leaders, to hear their stories and their encouragement to the younger generation, gave me a clear north star and set an extraordinarily high standard for what a PhD journey could be. I am deeply grateful to Edward and James for encouraging me to step outward into the world, and for their continued care, guidance, and support over the years.

Matus Telgarsky brought me to UIUC at a time when the application season had left me disheartened. Through dozens of emails, we discussed the challenges of machine learning theory and the motivations for pursuing theoretical research. Those exchanges restored a feeling of being understood and valued that I had not felt in a long time. I still remember him writing that he was "looking forward to the years ahead with me". Although our paths separated, I remain grateful for the light he brought into my early PhD years. I also thank him for his generosity and kindness as a friend, including moments that were deeply human,

such as watching cooking videos together during research meetings, when I started learning to live independently.

I am grateful to my advisor, Han Zhao, for taking me on after my collaboration with Matus concluded. This meant a great deal to me. Over the years, Han has given me unwavering support and, most importantly, remarkable freedom in research. This freedom allowed me to explore the questions I cared about most, to take intellectual risks, and to grow into an independent researcher with my own taste and character.

Although not my official advisor, Jiaqi Ma has been one of the most important mentors of my PhD. We began collaborating in Fall 2023 through Han's introduction and have since worked together on many projects for more than two years. Jiaqi brought me into the field of data attribution, and through our collaboration, we produced a fruitful line of research in this emerging area. Building on this foundation, I gradually shaped the broader agenda that became the core of my thesis, what I framed as the *science of data*. In addition, Jiaqi provided extensive mentorship on research and career decisions, and entrusted me with opportunities to mentor junior students.

I thank Arindam Banerjee, Nan Jiang, and Matus Telgarsky for serving on my qualifying exam committee, as well as Tong Zhang, David Forsyth, Chiyuan Zhang, and Sewoong Oh for serving on my thesis committee. I am especially grateful to Chiyuan, who has provided sustained mentorship and support since we began collaborating in Fall 2024, both in research and in career guidance. He has been a role model to me, and learning from him has been a privilege. I would also like to acknowledge the mentorship and support from Bin Cui, Bolin Ding, Jie Jiang, Bo Li, Feifei Li (Alibaba), Wei Liu, Dawn Song, Ye Ouyang, Jianan Zhang, Zheng Zhang, and James Zou.

In Summer 2025, I had the privilege of interning at Google Research, an experience that was both formative and unforgettable. I thank my hosts and mentors Zheng Xu, Ryan McKenna, Da Yu, and Peter Kairouz for helping me navigate Google, for their detailed feedback, and for their constant support. I am particularly thankful to Zheng for inviting

me to his home and for giving me the opportunity to co-organize an ICLR 2026 workshop, and to Peter for taking great care of me during the remainder of my internship. I also thank Daniel Ramage and Brendan McMahan for their generosity in taking the time to speak with me, and, together with Peter, for taking a chance on me by bringing me back to Google Research. I am excited to return after graduation to work on Gemini data and to begin the next chapter of my career.

Research is rarely a solitary endeavor, and I am grateful to the many collaborators who shaped my thinking, including Tianle Cai, Eli Chien, Junwei Deng, Gonzalo Munilla Garrido, Yifei He, Ziwei Ji, Nan Jiang, Ting-Wei Li, Yong Lin, Yunhui Long, Pritish Kamath, Rui Pan, Jiachen Wang, Shanshan Wu, Ruicheng Xian, Lydia Zakynthinou, and Jipeng Zhang. Beyond my own research projects, I found great fulfillment in contributing to the broader research community and supporting the growth of others. I thank Philip Amortila, Nathanael Assefa, Weixin Chen, Rohan Deb, and Cindy Zeng for co-organizing the UIUC Machine Learning Seminar with me, as well as Luxi He, Martin Jaggi, Ruoxi Jia, Pratyush Maini, Monica Ribero, Jiachen Wang, and Zheng Xu for co-organizing the 3rd DATA-FM workshop at ICLR 2026. I was also fortunate to mentor several talented junior students, including Pingbang Hu, Shixuan Liu, Weiyi Wang, and Qilong Wu. Watching their growth has been a constant source of joy.

I am also grateful to the support staff at the Siebel School of Computer Science and International Student & Scholar Services (ISSS) at the University of Illinois for their dedication and care. In particular, I thank Jennifer Comstock, Jessica Quicksall, Jennie Park, and Elizabeth Sheppard, whose behind-the-scenes work made it possible for me to focus on research and navigate the many administrative and logistical challenges of graduate school as an international student.

I am thankful for the friends who supported me throughout this journey. In particular, I thank the friends I made in the cornfield: Gargi Balasubramaniam, Wenxuan Bao, Qiuling Fan, Hongpeng Guo, Xiang Li, Zhaoheng Li, Yufei Ruan, Seiyun Shin, Haoxiang Wang,

Mingyuan Wu, Ruicheng Xian, Lang Yin, Minjian Zhang, and Yuanyi Zhong. They helped me navigate the campus and C-U, made me feel less isolated, and offered constant support. I am also grateful to many long-time and newly made friends: Yuanzhou Chen, Muthu Chidambaram, Jack Deschenes, Fangcheng Fu, Tianyu Guo, Luxi He, Baihe Huang, Putian Li, Licong Lin, Zhanran Lin, Ruoyang Liu, Yuhang Liu, Joe Melkonian, Qiuyu Ren, Shange Tang, Huiyuan Wang, Zhuofan Xie, Da Yu, Qingcheng Yu, Ruiqi Zhang, Yue Zhang, Xuyang Zhao, and Chufan Zheng. They brought warmth and joy into my life.

I thank JP and his family for their kindness and care, especially Gyung Youn (Ann) Baek, who looked after me when I first arrived at UIUC and helped me through an otherwise overwhelming time. I am also grateful for their warm hospitality during my trips to Las Vegas. I thank JL and David Faust for welcoming me into their home in New Jersey and for taking exceptional care of me when I was very ill in the winter of 2022. Finally, I thank Maria Carmen Domingo-Kirk for her generosity during my visits to Berkeley, for the many lunches she treated me to and the engaging conversations we shared, and for the spirit and energy she brings to life, which I have long admired.

I owe a unique debt of gratitude to Fan Wu. In research, she is the collaborator I trust unconditionally, the one who showed extraordinary patience toward my perfectionism even when it became excessive, and with whom I did my best work. But her support went far beyond this. I thank Fan for walking beside me through so many places and moments, across cities and seasons, and for standing by me during my darkest times. When I was most lost and exhausted, you stayed with me, helped me think and find a way forward.

I am deeply grateful to my parents and my family, especially my grandparents, who have been my most constant source of support throughout this journey. Conversations with them have been my most reliable way to step away from work, regain perspective, and recharge. Beyond encouragement and unconditional belief, my parents have offered something even more enduring: thoughtful perspective and steady guidance during difficult decisions. Their ability to help me reason clearly about complex choices, and to provide

concrete support when it mattered most, gave me a sense of stability and confidence that carried me through uncertainty. I am deeply aware that this has been a privilege, and I remain profoundly grateful for it.

Finally, as I did five years ago, I want to dedicate this last part to myself.

I thank myself for continuing through periods of doubt, when the meaning of this path was unclear and the costs felt tangible. I thank myself for choosing growth over certainty, for remaining willing to learn, revise, and begin again when early attempts failed, and for staying intellectually honest by resisting the temptation to simplify difficult questions or to mistake momentum for understanding.

Over time, I came to see research not only as a source of personal fulfillment, curiosity, beauty, and discovery, but as something whose meaning deepens when situated within a broader social context. Its value is realized most fully when it becomes a way to contribute, to support others, to build communities, and to leave behind a legacy that outlasts individual effort.

With this shift, my sense of self also changed. I learned to see myself not as the center of a narrative, but as one node within a larger network of mentors, collaborators, students, and communities. As my perspective widened, I became more aware of the privileges that enabled my path. I came to see privilege not as guilt or power, but as *leverage*: a position that carries responsibility to support others and to make the world, however modestly, better.

Most of all, I thank myself for learning to carry this responsibility with both clarity and warmth: to think carefully without becoming detached, to act without losing empathy, and to remain engaged with others and with the world as it is. This journey has taught me that becoming better as a researcher is inseparable from becoming better as a person, and that learning to transmit light is inseparable from having once been illuminated by it.

Contents

Chapter 1 Introduction 13
1.1 Thesis Overview 15
1.2 Bibliographic Notes 17
Chapter 2 Preliminaries and Background 18
2.1 Data Attribution 18
2.2 Differential Privacy 20
Chapter 3 A Unified Theory of Random Projection for Influence Functions . . . 21
3.1 Projection-Based Influence Approximation 26
3.1.1 Unregularized Projection 26
3.1.2 Regularized Projection 26
3.1.3 Factorized Influence 30
3.2 Influence with Out-of-Range Test Gradients 32
3.2.1 Leakage of Projection 33
3.2.2 Leakage of Factorized Influence 34
3.3 Experiment and Discussion 36
3.4 Related Work 39
3.5 Conclusion 40
Chapter 4 A Local Data Attribution Framework for Online Reinforcement Learning 41
4.1 Preliminaries 43
4.1.1 Online Reinforcement Learning 43
4.1.2 Data Attribution via TracIn 44
4.2 Framework Design 44
4.2.1 A Framework of Local Data Attribution 45
4.3 Applications of Local Data Attribution 48
4.3.1 Diagnosis of Learning: What Features Bottom Records? 49
4.3.2 Temporal Analysis of Behavior Formation: Phase Transition of Top Records 50
4.3.3 Targeted Interventions During Training: Filtering Amplifies Policy Gain 52
4.4 Iterative Influence-Based Filtering for Online RL Training 53
4.4.1 Algorithm and Designs 53

4.4.2 Experiments in Traditional RL Environments . . . 54
4.4.3 Extending IIF to RLHF for Large Language Models . . . 56
4.5 Related Work . . . 57
4.6 Conclusion . . . 58
Chapter 5 Empirical Privacy Variance . . . 59
5.1 Preliminaries . . . 61
5.2 Landscape of Empirical Privacy Variance . . . 61
5.2.1 Experimental Setups . . . 62
5.2.2 Trends and Generality of Empirical Privacy Variance . . . 65
5.2.3 Discussion . . . 66
5.3 How Hyperparameters Impact Empirical Privacy: Analysis and Selection Heuristics . . . 68
5.3.1 Dissecting Effects of Hyperparameters . . . 68
5.3.2 Improving Hyperparameter Selection . . . 71
5.3.3 Practical Evaluation of Proposed Heuristics . . . 73
5.4 Related Work . . . 75
5.5 Conclusion . . . 75
Chapter 6 Differentially Private Conditional Text Generation with RL-Boosted Control . . . 77
6.1 Preliminaries . . . 79
6.2 A Hierarchical Framework for DP Synthetic Text Generation . . . 80
6.2.1 A Hierarchical Framework . . . 80
6.2.2 Instantiation of the Framework . . . 82
6.2.3 Experiments . . . 83
6.3 Boosting Fine-Grained Control in ACTG with Anchored RL . . . 88
6.3.1 Measuring and Improving Instruction Following . . . 89
6.3.2 A Post-Training Recipe: Anchored RL . . . 90
6.3.3 Experiments . . . 92
6.4 Related Work . . . 92
6.5 Conclusion . . . 93
Chapter 7 Conclusion and Future Work . . . 94
7.1 Summary of Contributions . . . 94
7.2 Future Directions . . . 94

7.2.1 Technical Extensions . . . 95
7.2.2 Broader Themes . . . 96
A Appendix for Chapter 3: Theory of Random Projection . . . 98
A.1 Proofs for Sec. 3.1.1 (Unregularized Projection) . . . 98
A.2 Proofs for Sec. 3.1.2 (Regularized Projection) . . . 98
A.2.1 Proof of Resolvent Perturbation Concentration for Regularized Projection . . . 99
A.2.2 OSE-Based Alternative Analysis . . . 104
A.2.3 Proof of Anti-Concentration of Gaussian Sample Covariance . . . 106
A.2.4 Proof of Worst-Case Lower Bound . . . 111
A.3 Proofs for Sec. 3.1.3 (Factorized Influence) . . . 117
A.3.1 Proof of the Barrier of Unregularized Factorized Influence . . . 117
A.3.2 Proof of Factorized Resolvent Perturbation Concentration for Regularized Projection . . . 118
A.3.3 Note on Proof of Theorem 3.6 . . . 124
A.4 Proofs for Sec. 3.2.1 (Leakage of Projection) . . . 124
A.4.1 Proof Plan for Theorem 3.8 . . . 125
A.4.2 Proof of Single Test Gradient Leakage . . . 129
A.4.3 Proof of Multiple Test Gradients Leakage . . . 131
A.5 Proofs for Sec. 3.2.2 (Leakage of Factorized Influence) . . . 135
A.5.1 Proof Plan for Theorem 3.9 . . . 136
A.5.2 Proof of Concentration of Factor-Level Primitives . . . 140
B Appendix for Chapter 4: RL Data Attribution . . . 143
B.1 Detailed Experimental Setups . . . 143
B.1.1 Standard RL Environments . . . 143
B.1.2 Experimental Setups for Standard RL . . . 143
B.1.3 Experimental Setups for RLHF . . . 144
B.2 Additional Experimental Results . . . 145
B.2.1 More Demonstrations of Harmful Records . . . 145
B.2.2 Quantifying Phase Change via Weighted Graph Roughness Analysis 146
B.2.3 Additional Results for Single-Round Intervention . . . 149
B.2.4 Advantage-Based Heuristic . . . 150
B.2.5 TD Error Based Heuristic . . . 152

B.2.6 IIF Performance Under Various Filtering Percentages 154
B.2.7 Evaluating IIF with the Adam Optimizer 156
B.2.8 Statistical Significance of Final Performance Gains 157
B.2.9 Runtime for Experiments on Traditional RL Environments 157
B.2.10 Difficulty Based Heuristic 157
B.2.11 Comparing Two Target Functions for RLHF 158
B.2.12 A Breakdown of Runtime for the RLHF Experiments 158
C Appendix for Chapter 5: Empirical Privacy Variance 163
C.1 DP-SGD and DP-Adam 163
C.2 Additional Experimental Setups for Sec. 5.2 164
C.2.1 Enron Dataset Preprocessing Steps 164
C.2.2 TOFU Dataset Examples 165
C.2.3 TOFU Dataset Preprocessing Steps 165
C.2.4 Creating TOFU Dataset Variants 165
C.2.5 Verification of Fine-Tuning Data Exclusion from Pre-Training Corpora 168
C.2.6 Building the Secret Sets 169
C.2.7 Empirical Privacy Measures 173
C.2.8 Utility Measure 175
C.2.9 More Details of DP Fine-Tuning 175
C.3 Additional Experimental Results for Sec. 5.2 178
C.3.1 Additional Results on empirical privacy variance 178
C.3.2 Does Model Distance Explain the Trends of empirical privacy? . . . 180
C.3.3 Impact of Hyperparameters vs. Impact of Random Seeds 183
C.4 Additional Experimental Results for Sec. 5.3 184
C.4.1 Additional Results on Accuracy of Heuristics 184
C.4.2 Visualization of Selection Quality 184
C.4.3 Additional Results on Relative Privacy Risk 186
C.5 Results under the Worst-Case Privacy Measure 189
C.6 Complete Sets of Results 191
D Appendix for Chapter 6: ACTG-ARL 195
D.1 Additional Details of Our Approach 195
D.1.1 LLM-Assisted Schema Design and Extraction 195
D.1.2 Privacy Accounting 196

D.1.3 Details of the RL Algorithm PPO . . . 196
D.2 Full Algorithm and Pseudocode . . . 197
D.3 Additional Experimental Setups . . . 198
D.3.1 Datasets . . . 198
D.3.2 Implementation Details for the Hierarchical Framework and Baselines 199
D.3.3 A Comprehensive Evaluation Suite . . . 200
D.3.4 Implementation Details for ACTG-RL and ACTG-ARL . . . 202
D.4 Additional Experimental Results . . . 203
D.4.1 Issues with MAUVE Evaluation in the Literature . . . 203
D.4.2 Baseline of Single-Stage Conditional Text Generation . . . 204
D.4.3 Impact of Schema Richness . . . 205
D.4.4 Experiments on a Larger Model `gemma-3-4b-pt` . . . 206
D.4.5 Robustness to Oracle Choice . . . 207
D.4.6 Significance of Results . . . 207
D.4.7 Limitations of Direct Prompting as the Conditional Generator . . . 208
D.4.8 Aug-PE on PMC-Patients . . . 208
D.4.9 Aug-PE with a More Powerful Proprietary Model . . . 209
D.4.10 Failure Mode of CTCL . . . 209
D.4.11 Topic Distribution Matching . . . 210
D.4.12 Using IT Model for Conditional Generation . . . 210
D.4.13 Example of Reward Hacking . . . 212
D.4.14 Analysis on Best-of-N Data . . . 212
D.4.15 Additional Ablations on Anchored RL . . . 213
D.4.16 Utility Evaluation of Synthetic Data Produced by ACTG-ARL . . . 213
References . . . 227

Chapter 1 Introduction

Data has been a driving force behind many of the most transformative advances in artificial intelligence (AI). The breakthrough of AlexNet [1], often seen as the beginning of the modern deep learning era, was enabled by ImageNet [2], a large-scale dataset that also established a lasting benchmark for progress. In natural language processing, the availability of large pretraining corpora such as BookCorpus [3] and Common Crawl [4, 5] supported the development of early large-scale language models such as BERT [6] and GPT-1/2 [7], helping drive the shift from task-specific systems to general-purpose foundation models [8]. These examples illustrate a central theme: data has shaped AI not only by enabling new capabilities, but also by defining the benchmarks and regimes through which progress is measured.

The rise of generative AI has made the role of data even more pronounced. Scaling laws show that model performance improves systematically with increases in data, model size, and compute [9, 10]. Large pretraining corpora endow models with broad capabilities; curated instruction-response datasets teach them to follow human instructions [11, 12]; specialized corpora support domains such as mathematical reasoning and theorem proving [13]; and large-scale text-image datasets have driven major advances in generative vision models [14, 15]. Across these settings, progress depends not only on architecture and optimization, but increasingly on the scale, quality, and composition of the data itself.

At the same time, data has also become a central bottleneck and source of risk for the next generation of AI systems. High-quality web-scale corpora are becoming harder to obtain, while reliance on synthetic data raises concerns such as model collapse, where recursive training on model-generated content leads to degradation over time [16]. Privacy concerns are amplified when training data contains duplicated or user-contributed content, as shown by extraction attacks that can recover sensitive information, including personally identifiable information (PII) [17, 18]. Safety is likewise deeply tied to data, since harmful or biased training examples can propagate directly into model behavior [19, 20]. Meanwhile,

growing evidence suggests that quality often matters more than sheer quantity: carefully curated small datasets can match or even outperform much larger ones [21, 22, 23, 24]. Together, these developments suggest that continued progress in generative AI depends increasingly on understanding data itself: which data matters, how it should be used, and how it can be handled responsibly.

This thesis studies two complementary questions along this direction.

- **Data attribution.** How can we quantify the *value* of data in a principled and scalable way, and use this understanding to guide better data selection and filtering?
- **Data privacy.** How can we benefit from sensitive data without exposing it, and how should we reason about *privacy* in modern generative models?

Although these two directions may appear distinct, they share a deep conceptual connection rooted in a common question: *how does an individual training example influence a learned model?* Data attribution approaches this question from the perspective of *measurement*: it develops tools to quantify, for each training example, how much it contributes to a model's predictions. Differential privacy, by contrast, approaches it from the perspective of *control*: it designs training mechanisms whose outputs provably do not depend too strongly on any single data point. In other words, attribution seeks to *detect and measure* individual influence, while privacy seeks to *limit* it.

This duality suggests that the two directions are not merely parallel research threads, but complementary lenses on the same underlying phenomenon. Understanding one can inform the other: for example, attribution methods could serve as empirical probes for the effectiveness of privacy mechanisms, while privacy constraints may reshape the landscape of data influence in ways that attribution tools can reveal. Fig. 1 illustrates this relationship. We return to these connections in Chapter 7, where we discuss them as promising directions for future work.

The four papers in this thesis address these questions from different angles, but are

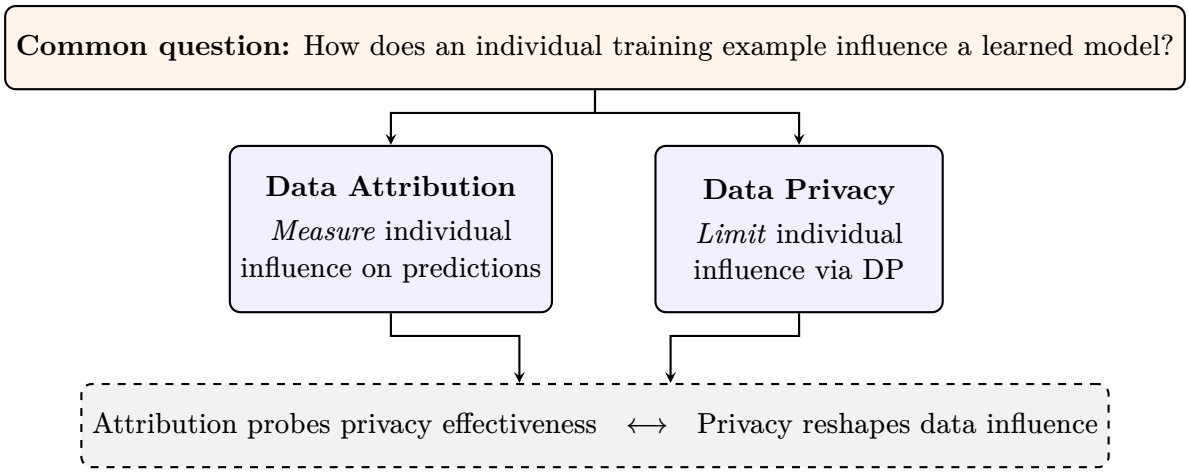


Figure 1: The two pillars of this thesis and their conceptual connection. Data attribution and data privacy address the same fundamental question from complementary perspectives: attribution *measures* individual data influence, while privacy *limits* it. Their interplay opens promising directions for future work (Chapter 7).

united by this common perspective: advancing generative AI requires elevating data from a passive input to a first-class object of study. Rather than viewing data merely as an i.i.d. sample in learning theory or as a fixed resource in model training, this thesis advocates a more principled treatment of data, one that seeks to quantify its value, optimize its use, safeguard its privacy, and, when necessary, convert sensitive data into useful synthetic data. In short, the thesis takes a data-centric perspective on generative AI.

1.1 Thesis Overview

This thesis is organized into two parts.

Part I: Data Attribution (Chapter 3–Chapter 4). The first part studies how to quantify the contribution of individual training examples to model behavior.

Chapter 3 develops a theoretical framework for random projection in influence functions, one of the most widely used tools for data attribution. Because influence scores depend on inverse curvature operators, their behavior under projection cannot be explained by standard Johnson–Lindenstrauss arguments alone. This chapter provides a unified analysis of when projection preserves influence in three settings: unregularized influence, ridge-regularized

influence, and Kronecker-factored influence. A central conclusion is that regularization changes the relevant complexity measure from rank to the *effective dimension* of the curvature matrix, yielding principled and instance-adaptive guidance for sketch size selection.

Chapter 4 extends data attribution to online reinforcement learning, with the goal of improving both interpretability and efficiency. Unlike supervised learning, online RL does not operate on a fixed dataset: the policy shapes the data, and the data in turn shapes the policy. To address this circular dependency, this chapter proposes a *local* attribution framework that analyzes one PPO training round at a time, attributing a model checkpoint update to the transition records in the recent rollout buffer. The resulting framework provides insight into learning dynamics and behavior formation, and also leads to a lightweight filtering algorithm that improves the efficiency and performance of online RL training.

Part II: Data Privacy (Chapter 5–Chapter 6). The second part studies how generative models can be trained or adapted on sensitive data under formal privacy constraints.

Chapter 5 introduces the notion of *empirical privacy variance*. The chapter shows that models trained with DP-SGD to satisfy the same formal (ε, δ)-DP guarantee can nonetheless exhibit substantially different empirical privacy behavior, as measured by memorization-based probes. This finding shows that the privacy budget alone does not determine practical privacy risk. The chapter further studies how hyperparameter choices shape this variation, identifies a trade-off between utility-oriented tuning and empirical privacy, and proposes heuristics for hyperparameter selection with empirical privacy taken into consideration.

Chapter 6 turns to a downstream objective of central importance: generating high-quality synthetic text under differential privacy. Rather than treating DP text generation as a monolithic problem, this chapter proposes a hierarchical framework that separates structured feature learning from conditional text generation. Building on this design, it

further introduces an anchored reinforcement learning method that improves controllability while mitigating reward hacking. Together, these components yield an end-to-end framework for differentially private conditional text generation that improves both generation quality and control.

1.2 Bibliographic Notes

The work presented in this thesis was conducted in collaboration with several researchers. Chapter 3 is based on [25] and is joint work with Pingbang Hu, Jiaqi Ma, and Han Zhao. Chapter 4 is based on [26] (NeurIPS 2025 oral) and is joint work with Fan Wu (equal contribution), Haotian Ye, David Forsyth, James Zou, Nan Jiang, Jiaqi Ma (equal advising), and Han Zhao (equal advising). Chapter 5 is based on [27] (ICML 2025) and is joint work with Fan Wu (equal contribution), Ruicheng Xian, Yuhang Liu, Lydia Zakynthinou, Pritish Kamath, Chiyuan Zhang, and David Forsyth. Chapter 6 is based on [28] (work conducted via interning at Google Research) and is joint work with Ryan McKenna, Da Yu, Shanshan Wu, Han Zhao, Zheng Xu, and Peter Kairouz.

This thesis does not include several other works on which I served as the primary author, mainly due to space constraints. For completeness, we list them here: a survey on data attribution [29]; a theoretical study on most influential subset selection [30] (NeurIPS 2024); a theoretical study on scalarization in multi-task learning [31] (NeurIPS 2023); a study on the trade-off between adversarial robustness and accuracy parity [32] (ICML 2023); and an SoK on privacy-preserving data synthesis [33] (IEEE S&P 2024).

Chapter 2 Preliminaries and Background

This chapter introduces the core concepts and technical tools that recur throughout the thesis. In particular, we review preliminaries on data attribution (Chapter 3–Chapter 4) and differential privacy (Chapter 5–Chapter 6). Additional background specific to each problem setting is deferred to the corresponding chapter.

2.1 Data Attribution

Data attribution aims to explain a trained model's behavior by tracing its predictions back to the training examples [29, 34]. Consider a model trained on a dataset $D = \{z_i\}_{i=1}^n$ by minimizing the empirical risk $\sum_{i=1}^n \ell(\theta, z_i)$. A data attribution method assigns each training example z_i a score that quantifies its contribution to a specified model behavior, such as the loss on a target test point. We focus on two representative methods. The first is *influence functions*, for which Chapter 3 develops theoretical foundations. The second is *TracIn*, a more practical and scalable tool adapted in Chapter 4.

Influence functions. Influence functions [35, 36] quantify how an infinitesimal perturbation of a training example affects the learned model and its predictions. Let

$$\hat{\theta} = \arg\min_{\theta} \sum_{i=1}^{n} \ell(\theta, z_i)$$

denote the empirical risk minimizer, and for a training point z, define

$$\hat{\theta}_{\varepsilon,z} = \arg\min_{\theta} \sum_{i=1}^{n} \ell(\theta, z_i) + \varepsilon\, \ell(\theta, z).$$

The influence of z on the loss at a test point z' is then defined as

$$\mathcal{I}(z, z') := \left.\frac{d\, \ell(\hat{\theta}_{\varepsilon,z}, z')}{d\varepsilon}\right|_{\varepsilon=0} = -\nabla_\theta \ell(\hat{\theta}, z')^\top H_{\hat{\theta}}^{-1} \nabla_\theta \ell(\hat{\theta}, z),$$

where

$$H_{\hat{\theta}} = \frac{1}{n} \sum_{i=1}^{n} \nabla_{\theta}^{2} \ell(\hat{\theta}, z_i)$$

is the Hessian of the empirical risk. In modern deep learning, $H_{\hat{\theta}}$ is often replaced by approximations such as the empirical Fisher matrix or the generalized Gauss–Newton matrix.

Applying influence functions to large-scale neural networks is computationally challenging due to the high dimensionality and the intractability of explicitly inverting the Hessian. A prominent line of work addresses this difficulty through *random projection*: gradients are compressed into a lower-dimensional space before influence scores are computed, often together with additional heuristics such as ridge regularization [37, 38, 39]. This practice is typically motivated by the Johnson–Lindenstrauss lemma. However, influence depends on an inverse-sensitive bilinear form, so standard distance-preservation guarantees do not directly imply preservation of influence scores. Establishing a rigorous theoretical understanding of this issue is the central goal of Chapter 3.

TracIn. TracIn [40] takes a different perspective by attributing model behavior through the entire *training trajectory*, rather than through a single trained model. It measures the cumulative effect of optimization steps involving a particular training example z_i on a target function $f(\theta)$. At iteration j, with parameters θ_j, learning rate η_j, and mini-batch $\mathcal{B}_j$, a first-order Taylor expansion yields

$$f(\theta_j) - f(\theta_{j+1}) \approx \eta_j \sum_{i \in \mathcal{B}_j} \nabla_{\theta} f(\theta_j) \cdot \nabla_{\theta} \ell(\theta_j, z_i).$$

Summing the contribution of z_i over the iterations in which it appears gives the TracIn score

$$\mathrm{TracIn}(z_i) = \sum_{j:\, z_i \in \mathcal{B}_j} \eta_j \, \nabla_{\theta} f(\theta_j) \cdot \nabla_{\theta} \ell(\theta_j, z_i).$$

Compared with influence functions, TracIn avoids Hessian inversion and relies only on gradient inner products, making it conceptually simpler and often more scalable in practice.

In Chapter 4, we adapt TracIn to the online reinforcement learning setting, where the training data are generated on the fly by an evolving policy.

2.2 Differential Privacy

Differential privacy (DP) [41] is a mathematical framework that limits the information an adversary can infer about any single training example from an algorithm's output. We say two datasets D and D' are *neighboring* if one can be obtained from the other by adding or removing a single sample.

Definition 2.1 ((ε, δ)-DP [41])**.** A randomized algorithm $\mathcal{M}$ is (ε, δ)-differentially private if for all neighboring datasets D, D' and for all $S \subseteq \text{Range}(\mathcal{M})$:

$$\Pr[\mathcal{M}(D) \in S] \leq e^{\varepsilon} \Pr[\mathcal{M}(D') \in S] + \delta.$$

Here, ε denotes the *privacy budget*, with smaller values indicating stronger privacy protection, and δ is a small failure probability. Together, (ε, δ) are referred to as the *privacy parameters*.

DP-SGD. DP-SGD [42] is the standard algorithm for training deep learning models with differential privacy guarantees. It modifies stochastic gradient descent by (1) clipping the per-sample gradient to bound its sensitivity, and (2) adding calibrated Gaussian noise. Formally, at step t, DP-SGD computes a *privatized gradient*:

$$\bar{g}_t := \frac{1}{|S_t|} \left(\sum_{z \in S_t} \frac{\nabla_\theta \ell(\theta_t, z)}{\max\left(1, \frac{\|\nabla_\theta \ell(\theta_t, z)\|}{c}\right)} + \mathcal{N}(0, \sigma^2 c^2 I) \right),$$

where S_t is a mini-batch of size b, c is the clipping norm, and the noise multiplier σ is determined by numerical privacy accountants [43, 44] to satisfy a target (ε, δ)-DP guarantee. The privatized gradient can also be used in other first-order optimizers such as Adam [45], leading to DP-Adam [46]. DP-SGD and its variants have been applied across diverse domains, including computer vision [47], language models [48], and federated learning [49].

Chapter 3 A Unified Theory of Random Projection for Influence Functions

Influence functions provide a principled approach to data attribution by quantifying how individual training examples affect a model's behavior [35, 36]. In modern neural networks, however, computing influence is prohibitively expensive: it requires manipulating extremely high-dimensional per-example gradients and inverting large, often ill-conditioned or singular, curvature operators F. To make influence estimation scalable, recent methods rely on *random projection*, compressing gradients and curvature into a much lower-dimensional space before performing the attribution computation [37, 38, 39, 50].

Despite its empirical success, a theoretical understanding of this practice is still lacking. Existing works often appeal heuristically to the Johnson–Lindenstrauss (JL) lemma [51], since standard sketches such as Gaussian, Rademacher, and sparse JL maps approximately preserve Euclidean geometry [52, 53, 54, 55]. However, influence is governed not by ordinary Euclidean distances, but by an *inverse-sensitive* bilinear form involving F^{-1}. As a result, classical JL guarantees do not directly explain when projection preserves influence scores. More broadly, prior theory offers limited guidance on how the required sketch dimension depends on the problem geometry or on additional design choices such as regularization, even though both are empirically important [56].

This chapter develops a unified theory of random projection for influence functions. Our analysis characterizes the sketch dimension required to provably preserve influence scores and clarifies how this requirement depends on the geometry of the underlying curvature and gradient spaces. We further show how the same framework applies to three widely used large-scale variants: *unregularized projection* [37, 38], *regularized projection* [57, 58], and *Kronecker-factored influence* [39, 50], which combines factorized projection with structured curvature approximations such as K-FAC [59, 60].

Setup and Notation. Let g and g' denote training and test gradients with respect to the trained model parameters $\theta \in \mathbb{R}^d$, and let $F \succeq 0$ be a curvature matrix evaluated at θ,

with $r := \operatorname{rank}(F)$. Typical choices of F include the generalized Gauss–Newton matrix [58, 61] and the empirical Fisher $\frac{1}{n}\sum_{i=1}^{n} g_i g_i^\top$ [62, 63], both standard approximations to the Hessian. We study the inverse-sensitive bilinear form with a ridge parameter $\lambda \geq 0$, denoted as $\tau_\lambda(g, g') := g^\top (F + \lambda I_d)^{-1} g'$, where F^{-1} denotes either the matrix inverse or the Moore–Penrose pseudoinverse when F is singular. Unless otherwise stated, we let $P \in \mathbb{R}^{m \times d}$ denote a sketch whose rows are i.i.d. $1/\sqrt{m}$-scaled isotropic sub-Gaussian vectors [64, Chapter 2].[1] Such matrices are commonly referred to as *oblivious sketching matrices* and include Gaussian, Rademacher, and sparse JL transforms widely used in practice. The resulting projected (possibly regularized) influence is defined $\tilde{\tau}_\lambda(g, g') := (Pg)^\top (PFP^\top + \lambda I_m)^{-1} (Pg')$.

Our Contributions. We present a sequence of results characterizing when projection provably preserves influence functions across a range of settings. Under the assumption $g, g' \in \operatorname{range}(F)$, we precisely delineate when projection *can* and *cannot* succeed without regularization, show how ridge regularization alters the required sketch size, and extend the analysis to Kronecker-factored curvature approximations. We then relax the assumption on g' and quantify an additional sketch-induced *leakage* term arising from components of the test gradient in $\ker(F)$, yielding guarantees for influence queries at general, unseen test points.

First, we ask whether sketching can preserve the unregularized influence $\tau_0(g, g')$. We show a dichotomy: unless the sketch is injective on $\operatorname{range}(F)$, uniform multiplicative approximation is impossible, in the sense that no bound of the form $|\tau_0(g, g') - \tilde{\tau}_0(g, g')| \leq \varepsilon \tau_0(g, g')$ can hold for all g, g' and any $\varepsilon > 0$. Conversely, injectivity on $\operatorname{range}(F)$ guarantees *exact* preservation.

[1] A mean-zero random variable X is *sub-Gaussian* with parameter σ^2 if $\mathbb{E}[\exp(tX)] \leq \exp(\sigma^2 t^2/2)$ for all $t \in \mathbb{R}$; a random vector is sub-Gaussian if all one-dimensional marginals are sub-Gaussian.

Main Result 1 (Unregularized projection, Theorem 3.1): Let $F \succeq 0$ with $r := \operatorname{rank}(F)$. For $\lambda = 0$, for all $g, g' \in \operatorname{range}(F)$, $\tilde{\tau}_0(g, g') = \tau_0(g, g')$ *if and only if* P is injective on $\operatorname{range}(F)$. If P is not injective on $\operatorname{range}(F)$ (in particular if $m < r$), then for any constant factor, no uniform multiplicative approximation guarantee is possible over $g, g' \in \operatorname{range}(F) \setminus \{0\}$.

Theorem 3.1 shows that, without regularization, influence preservation requires m to scale on the order of r. In contrast, when ridge regularization is employed, we show that the required sketch size is no longer governed by r but instead by the *effective dimension* $d_\lambda(F) := \operatorname{tr}(F(F + \lambda I)^{-1})$, a classical notion in Bayesian model selection [65, 66]. This quantity is always bounded above by r and can be substantially smaller when the spectrum of F decays quickly.

Main Result 2 (Regularized projection: Theorems 3.2 and 3.4): Fix $\lambda > 0$ and define $d_\lambda(F) = \operatorname{tr}(F(F + \lambda I)^{-1})$. If $m = \Omega((d_\lambda(F) + \log(1/\delta))/\varepsilon^2)$, then with probability at least $1 - \delta$, for all $g, g' \in \operatorname{range}(F)$,

$$\left|\tilde{\tau}_\lambda(g, g') - \tau_\lambda(g, g')\right| \leq \varepsilon \sqrt{\tau_0(g, g)} \sqrt{\tau_0(g', g')}$$

Conversely, for Gaussian oblivious sketches, there exist $F \succeq 0$ such that if $m = o(d_\lambda(F)/\varepsilon^2)$, there exists some $g, g' \in \operatorname{range}(F)$ admits an $\Omega(\varepsilon)$ error with constant probability.

In large neural networks, influence computation hinges on curvature inversion, yet forming or inverting the full empirical Hessian or Fisher is infeasible. Consequently, practical pipelines adopt structured curvature approximations, most notably Kronecker-factored approximate curvature (K-FAC). This motivates us to develop a projection theory tailored to this setting.

As reviewed in Sec. 3.1.3, K-FAC models the curvature as $F = A \otimes E$ (in a layerwise

manner), where A and E capture the empirical covariances of forward activations and backpropagated gradients, respectively. To exploit this structure, one natural idea is to enforce the sketch to share the same factorization $P = P_A \otimes P_E$, where P_A and P_E are oblivious sketching matrices [39]. While this yields substantial computational savings, the Kronecker structure breaks the i.i.d. row assumption on P, rendering a direct adaptation of Theorems 3.1 and 3.2 inapplicable. We overcome this technical challenge through a fine-grained analysis and establish rigorous approximation guarantees.

Main Result 3 (Factorized influence, Theorems 3.5 and 3.6): Assume $F = A \otimes E \succeq 0$ and a Kronecker sketch $P = P_A \otimes P_E$ with factor sketch sizes m_A and m_E.

(i) **Unregularized barrier.** For $\lambda = 0$, exact invariance on range(F) holds if and only if P_A is injective on range(A) and P_E is injective on range(E), which in particular necessitates $m_A \geq \text{rank}(A)$ and $m_E \geq \text{rank}(E)$.

(ii) **Regularized approximation.** Let P_A and P_E each to be oblivious sketch. For $\lambda > 0$, letting $\lambda_E := \lambda/\|E\|_2$ and $\lambda_A := \lambda/\|A\|_2$, if $m_A = \Omega((d_{\lambda_E}(A) + \log(1/\delta))/\varepsilon^2)$ and $m_E = \Omega((d_{\lambda_A}(E) + \log(1/\delta))/\varepsilon^2)$, then with probability at least $1 - \delta$, for all $g, g' \in \text{range}(F)$,

$$|\tilde{\tau}_\lambda(g, g') - \tau_\lambda(g, g')| \leq \varepsilon \sqrt{\tau_0(g, g)} \sqrt{\tau_0(g', g')}.$$

Finally, we note that all of the above guarantees are stated for gradients lying in range(F), which, when F is the empirical Fisher, includes all training gradients used for attribution. In practice, however, a test gradient g' may have a component in $\ker(F)$. We show that, in both the unregularized and regularized settings, these components do not affect the true (unsketched) influence, while sketching introduces an additional *leakage* term. We quantify this "out-of-range leakage" and show it decays at the usual $O(m^{-1/2})$ rate with explicit dependence on λ and the spectrum of F.

Main Result 4 (Projection leakage, Theorems 3.8 and 3.9): For a general $g' \in \mathbb{R}^d$, write $g' = g'_{\parallel} + g'_{\perp}$ with $g'_{\parallel} \in \text{range}(F)$ and $g'_{\perp} \in \ker(F)$. We show that in this case, for either $\lambda = 0$ or $\lambda > 0$,

$$|\tilde{\tau}_\lambda(g, g') - \tau_\lambda(g, g')| \leq |\tilde{\tau}_\lambda(g, g'_{\parallel}) - \tau_\lambda(g, g'_{\parallel})| + |\tilde{\tau}_\lambda(g, g'_{\perp})|,$$

with an additional leakage error $|\tilde{\tau}_\lambda(g, g'_{\perp})|$ beyond Theorem 3.2. We then prove in Theorem 3.8 that for a collection of k test gradients $\{g'_j\}_{j=1}^k$, with sketch size $m = \Omega\big((r + \log(k/\delta))/\varepsilon^2\big)$,[a]

(i) **Unregularized**: $|\tilde{\tau}_0(g, g'_{\perp})| \leq \varepsilon \|g\|_2 \|g'_{\perp}\|_2 / \lambda^+_{\min}(F)$.

(ii) **Regularized**: $|\tilde{\tau}_\lambda(g, g'_{\perp})| \leq \varepsilon \|g\|_2 \|g'_{\perp}\|_2 (1/\lambda + 2\|F\|_2/\lambda^2)$.

Moreover, in Theorem 3.9, we show that similar leakage guarantees extend to the factorized influence setting.

[a] Or alternatively linear in $k' = \dim(\text{span}(\{g'_{j,\perp}\}_{j=1}^k))$, which in practice is usually worse than $\log(k)$.

Taken together, we develop a unified theory for when projection can provably approximate influence-style data attribution scores of the form $g^\top (F + \lambda I)^{-1} g'$. Specifically, without regularization, projection preserves influence for all $g, g' \in \text{range}(F)$ only when the sketch is injective on $\text{range}(F)$, which essentially forces $m \geq \text{rank}(F)$; otherwise, uniform multiplicative approximation is impossible. With regularization, the required sketch size is instead governed by the effective dimension $d_\lambda(F)$. We further extend these guarantees to Kronecker-factored (K-FAC-style) curvature and sketches. Finally, we quantify an additional sketch-induced leakage term that can appear when test gradients have components in $\ker(F)$. Overall, our results provide principled, instance-adaptive guidance for choosing m and clarify how projection interacts with regularization and structured curvature approximations.

3.1 Projection-Based Influence Approximation

3.1.1 Unregularized Projection

In this section, we show that in the absence of regularization, projection alone encounters a fundamental barrier in the sketch size m. In particular, there is a sharp phase transition: when $m < r$, no multiplicative approximation guarantee is possible; whereas when $m \geq r$, a continuous sketch yields exact invariance with probability one.

Theorem 3.1 (Barrier of unregularized projection)**.** *The equality $\tau_0(g, g') = \tilde{\tau}_0(g, g')$ holds for any $g, g' \in \text{range}(F)$* iff *$P$ is injective on* $\text{range}(F)$*, i.e.* $\text{rank}(PU) = \text{rank}(F) = r$ *where $F = U\Lambda U^\top$ is the compact eigendecomposition of F with $U \in \mathbb{R}^{d\times r}$ orthonormal and $\Lambda \in \mathbb{R}^{r\times r}$ positive definite. Subsequently, for any PSD $F \in \mathbb{R}^{d\times d}$ and* ***any*** *matrix $P \in \mathbb{R}^{m\times d}$, one cannot hope to obtain any multiplicative approximation of $\tau_0(g, g')$ via $\tilde{\tau}_0(g, g')$ when* $\text{rank}(PU) < r$.

The proof can be found in Appendix A.1. The key intuition is that, without regularization, influence depends on exact inversion over $\text{range}(F)$. Any collapse of directions within $\text{range}(F)$ renders F^{-1} ill-defined after sketching, hence no multiplicative control is possible. Consequently, exact preservation of unregularized influence requires the sketch to be injective on $\text{range}(F)$, which in turn forces $m \geq r$. In overparameterized regimes where high-dimensional per-sample gradients are in general position, one typically has $r \approx n$, and thus m must scale with the dataset size. In contrast, we will show that introducing ridge regularization ($\lambda > 0$) fundamentally changes this requirement, with the sketch size governed by the effective dimension $d_\lambda(F)$, which can be substantially smaller than r.

3.1.2 Regularized Projection

Unlike the unregularized case, in this section, we show that for the projected influence function, the extra damping term λI_d helps control the effective dimension by shrinking small eigenvalues of the curvature operator F, effectively reducing the Gaussian complexity

governing the uniform concentration bound. In particular, we show that the sketch size requires *only* to scale with the *effective dimension* of F with $\lambda > 0$:

$$d_\lambda(F) := \operatorname{tr}\left(F(F+\lambda I_d)^{-1}\right) = \sum_{j=1}^{r} \frac{\lambda_j(F)}{\lambda_j(F)+\lambda} \le r.$$

In practice, as we shall observe in Sec. 3.3, the spectrum of F decays rapidly, and thus $d_\lambda \ll r \ll d$ for moderate λ. Hence, requiring the sketch size m to scale only with the effective dimension d_λ at scale λ, rather than the ambient dimension d or the rank r of F, makes the regularized projection approach feasible at scale. We now state the theorem and sketch the proof below.

Theorem 3.2 (Upper bound of regularized projection)**.** *Let $P \in \mathbb{R}^{m\times d}$ be an oblivious sketching matrix with rows $P_i^\top = \frac{1}{\sqrt{m}} W_i^\top$, where $\{W_i\}_{i=1}^m \sim W$ are i.i.d. sub-Gaussian random vectors in $\mathbb{R}^d$ satisfying $\mathbb{E}[W] = 0$ and $\mathbb{E}[WW^\top] = I_d$.*[2] *For any $\varepsilon, \delta \in (0,1)$, if the sketch size satisfies*

$$m = \Omega\left(\frac{d_\lambda(F) + \log(1/\delta)}{\varepsilon^2}\right),$$

then with probability at least $1-\delta$, the following bounds hold for all $g, g' \in \operatorname{range}(F)$:

$$|\tilde{\tau}_\lambda(g,g') - \tau_\lambda(g,g')| \le \varepsilon \sqrt{\tau_0(g,g)}\sqrt{\tau_0(g',g')}.$$

Proof. Let $g, g' \in \operatorname{range}(F)$ and write $g = F^{1/2}y$ and $g' = F^{1/2}y'$. Using the push-through identity $A(A^\top A + \lambda I)^{-1} = (AA^\top + \lambda I)^{-1}A$ with $A = PF^{1/2}$, and defining $G := F^{1/2}P^\top PF^{1/2}$ yields

$$\begin{aligned}\tilde{\tau}_\lambda(g,g') &= (Pg)^\top (PFP^\top + \lambda I)^{-1}(Pg') \\ &= y^\top F^{1/2}P^\top (PFP^\top + \lambda I)^{-1} PF^{1/2}y' = y^\top G(G+\lambda I)^{-1}y'.\end{aligned}$$

[2]Since we only assume bounded sub-Gaussian norm on the random vectors W_i, the result applies to a wide range of random projection matrices, including Gaussian, Rademacher, and sparse JL transform.

On the other hand, define $B := F^{1/2}(F+\lambda I)^{-1/2}$ as the *λ-whitened influence subspace.* Since F and $F+\lambda I$ are simultaneously diagonalizable (they share the eigenbasis of F), all matrix functions of these operators commute; in particular, $F^{1/2}$, $(F+\lambda I)^{-1/2}$, and $(F+\lambda I)^{-1}$ commute and $BB^\top = B^\top B = B^2 = F(F+\lambda I)^{-1}$. Hence, we have

$$\tau_\lambda(g,g') = g^\top (F+\lambda I)^{-1} g' = y^\top BB^\top y' = y^\top F(F+\lambda I)^{-1} y',$$

which gives $|\tilde{\tau}_\lambda(g,g') - \tau_\lambda(g,g')| = |y^\top G(G+\lambda I)^{-1}y' - y^\top F(F+\lambda I)^{-1}y'|$. Thus, it suffices to control the operator norm of $F(F+\lambda I)^{-1} - G(G+\lambda I)^{-1}$.

Note $\|B\|_2 \le 1$. Applying Theorem 1.2 with $M = B$ and $m = \Omega(\varepsilon^{-2}(d_\lambda(F) + \log(1/\delta)))$ yields $\|B^\top(P^\top P - I)B\|_2 \le \varepsilon/2$. Conjugating by $(F+\lambda I)^{1/2}$ and using $G = F^{1/2}P^\top P F^{1/2}$, this implies a PSD sandwich

$$\left(1 - \tfrac{\varepsilon}{2}\right)(F+\lambda I) \preceq (G+\lambda I) \preceq \left(1 + \tfrac{\varepsilon}{2}\right)(F+\lambda I).$$

Inverting the sandwich gives $\|(G+\lambda I)^{-1} - (F+\lambda I)^{-1}\|_2 \le \frac{1}{\lambda} \cdot \frac{\varepsilon/2}{1-\varepsilon/2} \le \varepsilon/\lambda$. Finally, using the identity $A(A+\lambda I)^{-1} = I - \lambda(A+\lambda I)^{-1}$ for any PSD A, we get

$$\left\|F(F+\lambda I)^{-1} - G(G+\lambda I)^{-1}\right\|_2 = \lambda \left\|(G+\lambda I)^{-1} - (F+\lambda I)^{-1}\right\|_2 \le \varepsilon,$$

which is the desired operator control (formal details are in Theorem 1.3). Therefore,

$$|\tilde{\tau}_\lambda(g,g') - \tau_\lambda(g,g')| = \left|y^\top \left[G(G+\lambda I)^{-1} - F(F+\lambda I)^{-1}\right] y'\right| \le \varepsilon \|y\|_2 \|y'\|_2.$$

As $\|y\|_2^2 = \tau_0(g,g)$ and $\|y'\|_2^2 = \tau_0(g',g')$, we conclude the proof. □

Remark 3.3. The core technical challenge in the proof of Theorem 3.2 is to bound $\|F(F+\lambda I)^{-1} - G(G+\lambda I)^{-1}\|_2$. A natural alternative is to invoke an oblivious subspace embedding

(OSE) [67]. For a fixed matrix $A \in \mathbb{R}^{d\times r}$, $P \in \mathbb{R}^{m\times d}$ is an ε-OSE for range(A) if

$$-\varepsilon A^\top A \preceq A^\top (P^\top P - I)A \preceq \varepsilon A^\top A.$$

Instantiating $A = F^{1/2}$ yields a sandwich $(1-\varepsilon)F \preceq G \preceq (1+\varepsilon)F$, which implies $\|F(F+\lambda I)^{-1} - G(G+\lambda I)^{-1}\|_2 = O(\varepsilon)$ by operator monotonicity of $t \mapsto t/(t+\lambda)$ (see Appendix A.2.2). However, OSE enforces uniform multiplicative accuracy over range$(F^{1/2})$, so even directions with $\lambda_j(F) \ll \lambda$ must be preserved up to a $(1\pm\varepsilon)$ factor, leading to $m = \Omega(r/\varepsilon^2)$ [67, Theorems 2.3 and 6.10].

Our proof instead exploits the weaker, λ-dependent requirement: it suffices for P to be an approximate isometry on the λ-whitened influence subspace $B = F^{1/2}(F+\lambda I)^{-1/2}$, i.e., $\|B^\top(P^\top P - I)B\|_2 \le O(\varepsilon)$. This yields the **ridge-regularized sandwich** $(1-\varepsilon)(F+\lambda I) \preceq G + \lambda I \preceq (1+\varepsilon)(F+\lambda I)$. Importantly, this condition controls $F+\lambda I$ rather than F itself: in directions where $\lambda_j(F) \ll \lambda$, both $F+\lambda I$ and $G+\lambda I$ are dominated by λ, so even large relative errors in F have negligible impact on the inverse. Consequently, such low-eigenvalue directions need not be preserved multiplicatively, and the required sketch size is governed by the effective dimension at scale λ, yielding the sharper bound $m = \Omega(d_\lambda(F)/\varepsilon^2)$.

We now complement Theorem 3.2 with a worst-case matching lower bound, showing that the effective dimension $d_\lambda(F)$ characterizes the tight dependence of m for oblivious sketching in regularized influence. Concretely, we show that for Gaussian oblivious sketches, if the sketch size is smaller than $\Theta(d_\lambda(F)/\varepsilon^2)$, then there exist problem instances on which the sketched influence incurs $\Omega(\varepsilon)$ error with constant probability.

Theorem 3.4 (Lower bound for regularized projection)**.** *Let $P \in \mathbb{R}^{m\times d}$ be a Gaussian oblivious sketch with rows i.i.d. $\mathcal{N}(0, I_d)$. There exists a family of $F \in \mathbb{R}^{d\times d}$ such that if $m = o(d_\lambda(F)/\varepsilon^2)$, then there exists $g \in$ range(F) with $|\tilde{\tau}_\lambda(g,g) - \tau_\lambda(g,g)| = \Omega(\varepsilon)\tau_0(g,g)$ with constant probability.*

Full details are in Appendix A.2. We see that Theorem 3.4 formalizes a worst-case

limitation for this class of sketches: with Gaussian oblivious projections, one cannot uniformly beat the $d_\lambda(F)/\varepsilon^2$ scaling. Combined with the instance-adaptive upper bound in Theorem 3.2, this identifies $d_\lambda(F)$ as the fundamental complexity parameter governing regularized projection.

3.1.3 Factorized Influence

In many large-scale settings, explicitly forming or inverting the empirical Fisher/Hessian F is infeasible, and second-order methods instead rely on structured approximations. A common choice is a Kronecker factorization (e.g., K-FAC [59, 62]), which models each layerwise block as $F \approx A \otimes E$ for smaller PSD factors $A \in \mathbb{R}^{d_A \times d_A}$ and $E \in \mathbb{R}^{d_E \times d_E}$, which are forward activation and backprop-gradient covariances, respectively.

This structure suggests a natural computational counterpart on the sketching side: use a *factorized sketch* $P = P_A \otimes P_E$, where $P_A \in \mathbb{R}^{m_A \times d_A}$ and $P_E \in \mathbb{R}^{m_E \times d_E}$ are respectively the standard oblivious sketching considered in Theorem 3.2.[3] The resulting sketch has ambient dimension $d := d_A d_E$ and sketch dimension $m := m_A m_E$, i.e., $P \in \mathbb{R}^{m \times d}$. Moreover, write a per-example layer gradient as a matrix $G \in \mathbb{R}^{d_E \times d_A}$ with $g = \mathrm{vec}(G) \in \mathbb{R}^d$. Then the projection can be computed without materializing the full $m \times d$ sketching matrix as $Pg = (P_A \otimes P_E)\,\mathrm{vec}(G) = \mathrm{vec}(P_E G P_A^\top)$. Consequently, the per-example cost reduces to two smaller multiplies $P_E G$ and $(P_E G)P_A^\top$, plus solving the resulting regularized system in sketch dimension m. Similarly, we can also form the sketched curvature efficiently: using the mixed-product identity of Kronecker products, $PFP^\top = (P_A \otimes P_E)(A \otimes E)(P_A \otimes P_E)^\top = (P_A A P_A^\top) \otimes (P_E E P_E^\top)$.

In the unregularized case ($\lambda = 0$), the exact invariance barrier becomes strictly more stringent under a Kronecker sketch: exact preservation on $\mathrm{range}(F)$ holds if and only if *both* factor sketches are injective on their respective ranges.

[3] Concretely, rows of P_A and P_E are i.i.d. isotropic sub-Gaussian random vectors with scaling $1/\sqrt{m_A}$ or $1/\sqrt{m_E}$.

Theorem 3.5 (Barrier of unregularized projection for factorized influence)**.** *Let* $F = A \otimes E \succeq 0$ *and* $P = P_A \otimes P_E$ *as above. Then* $\tilde{\tau}_0(g, g') = \tau_0(g, g')$ *for all* $g, g' \in \mathrm{range}(F)$ *if and only if* P_A *is injective on* $\mathrm{range}(A)$ *and* P_E *is injective on* $\mathrm{range}(E)$*. In particular, this necessitates* $m_A \geq \mathrm{rank}(A)$ *and* $m_E \geq \mathrm{rank}(E)$*, hence* $m = m_A m_E \geq \mathrm{rank}(A)\,\mathrm{rank}(E) = \mathrm{rank}(F)$*.*

See Appendix A.3.1 for a proof. This motivates integrating regularization and considering

$$\begin{aligned}\tilde{\tau}_\lambda(g, g') &= (Pg)^\top (PFP^\top + \lambda I_m)^{-1} (Pg') \\ &= \mathrm{vec}(P_E G P_A^\top)^\top \left((P_A A P_A^\top) \otimes (P_E E P_E^\top) + \lambda I_m\right)^{-1} \mathrm{vec}(P_E G' P_A^\top).\end{aligned}$$

However, factorization changes the sketching analysis: when $P = P_A \otimes P_E$, the matrix $P^\top P$ is no longer a standard i.i.d. sample covariance, so the covariance-type deviation driving the proof of Theorem 3.2 requires a dedicated argument. We now present the corresponding approximation guarantee for regularized projection under this factorized model. The key technical step is a factorized covariance deviation bound (Theorem 1.5), proved in Appendix A.3.

Theorem 3.6 (Upper bound of regularized projection for factorized influence)**.** *Let* $F = A \otimes E \succeq 0$ *and* $P = P_A \otimes P_E$ *be as above, with the factors* P_A, P_E *denoting the sketching matrices defined in Theorem 3.2. Assume* $\lambda \leq \|A\|_2 \|E\|_2$*, and define the rescaled regularization levels* $\lambda_E := \lambda / \|E\|_2$ *and* $\lambda_A := \lambda / \|A\|_2$*. For any* $\varepsilon, \delta \in (0, 1)$*, if the sketch sizes for* P_A *and* P_E *satisfy*

$$m_A = \Omega\left(\frac{d_{\lambda_E}(A) + \log(1/\delta)}{\varepsilon^2}\right), \qquad m_E = \Omega\left(\frac{d_{\lambda_A}(E) + \log(1/\delta)}{\varepsilon^2}\right),$$

then with probability at least $1 - \delta$*, the following holds for all* $g, g' \in \mathrm{range}(F)$*:*

$$|\tilde{\tau}_\lambda(g, g') - \tau_\lambda(g, g')| \leq \varepsilon \sqrt{\tau_0(g, g)} \sqrt{\tau_0(g', g')}.$$

Proof. The proof follows the same template as Theorem 3.2. Let $B := F^{1/2}(F + \lambda I)^{-1/2}$

and $G \coloneqq F^{1/2}P^\top PF^{1/2}$, and the key step is again to control the covariance-type deviation $\|B^\top(P^\top P - I)B\|_2$. We apply Theorem 1.5 (proved in Appendix A.3) with parameters $\varepsilon_0 \coloneqq \varepsilon/10$ and $\delta_0 \coloneqq \delta/2$. Under the stated conditions on m_A and m_E, this yields that with probability at least $1 - 2\delta_0 = 1 - \delta$,

$$\|B^\top(P^\top P - I)B\|_2 \le 2\varepsilon_0 + 3\varepsilon_0^2 \le \varepsilon/2,$$

where the last inequality uses $\varepsilon \in (0, 1)$. On this event, the same PSD sandwich and resolvent perturbation argument used in Theorem 1.3 implies $\|F(F + \lambda I)^{-1} - G(G + \lambda I)^{-1}\|_2 \le \varepsilon$, which in turn gives the stated bilinear (and quadratic) influence error bounds. □

Remark 3.7. Theorem 3.6 highlights a fundamental computational–statistical trade-off. While factorized sketches offer substantial computational and memory advantages over unfactorized ones, they incur a higher statistical cost in terms of the required sketch size. In particular, since the total sketch size is $m = m_A m_E$, achieving an ε-approximation error requires $m = m_A m_E = \widetilde{\Omega}(\varepsilon^{-4}(d_{\lambda_E}(A)d_{\lambda_A}(E)))$, which exhibits a worse dependence on ε (from ε^{-2} to ε^{-4}) compared to the unfactorized sketch guarantee in Theorem 3.2. Importantly, this gap is not an artifact of loose analysis, but follows from the separable nature of the factorized sketch: P_A and P_E must independently satisfy an ε-level concentration bound at its own regularization scale. Consequently, the total sketch size reflects the product of the factor-level requirements. As a result, factorized sketches are most effective in regimes where the computational and memory savings from separability outweigh the increased statistical overhead.

3.2 Influence with Out-of-Range Test Gradients

The analysis in Sec. 3.1 assumes that both arguments of the (regularized) influence bilinear form lie in range(F). This assumption is natural for training gradients: when F is instantiated as the (empirical) Fisher information matrix, $F = \frac{1}{n}\sum_{i=1}^n g_i g_i^\top$, every training

gradient lies in range(F) by construction. In practice, however, we are often interested in the influence of an unseen test point z' with respect to a training point z, for which the corresponding test gradients g' need not lie in range(F).

We extend the above guarantees to this setting by explicitly characterizing the additional sketch-induced error arising from the component of g' orthogonal to range(F).

3.2.1 Leakage of Projection

To make the source of this additional term explicit, we decompose $g' = g'_{\|} + g'_{\perp}$, where $g'_{\|} \in \text{range}(F)$ and $g'_{\perp} \in \ker(F)$, such that the decomposition is orthogonal in the Euclidean inner product. Using linearity of $\tau_\lambda(\cdot,\cdot)$ and $\tilde{\tau}_\lambda(\cdot,\cdot)$ in their second argument, we have $\tau_\lambda(g,g') = \tau_\lambda(g,g'_{\|}) + \tau_\lambda(g,g'_{\perp})$ and $\tilde{\tau}_\lambda(g,g') = \tilde{\tau}_\lambda(g,g'_{\|}) + \tilde{\tau}_\lambda(g,g'_{\perp})$. Consequently,

$$|\tilde{\tau}_\lambda(g,g') - \tau_\lambda(g,g')| = \left|(\tilde{\tau}_\lambda(g,g'_{\|}) - \tau_\lambda(g,g'_{\|})) + \tilde{\tau}_\lambda(g,g'_{\perp}) - \tau_\lambda(g,g'_{\perp})\right|.$$

Observe that the true (regularized) influence does not couple range(F) and $\ker(F)$, i.e., $\tau_\lambda(g,g'_{\perp}) = g^\top(F+\lambda I)^{-1}g'_{\perp} = 0$ for all $\lambda \geq 0$: indeed, F and $(F+\lambda I)^{-1}$ share the same eigenbasis, and since $g \in \text{range}(F)$ and $g'_{\perp} \in \ker(F)$, hence $(F+\lambda I)^{-1}g$ and $g'_{\perp}$ lie in orthogonal subspaces. Hence,

$$|\tilde{\tau}_\lambda(g,g') - \tau_\lambda(g,g')| \leq |\tilde{\tau}_\lambda(g,g'_{\|}) - \tau_\lambda(g,g'_{\|})| + |\tilde{\tau}_\lambda(g,g'_{\perp})|.$$

The first term can be bounded via Theorem 3.2; on the other hand, the remaining term is a purely sketch-induced artifact: the sketch can introduce a nonzero *leakage term* $\tilde{\tau}_\lambda(g,g'_{\perp})$ due to mixing between range(F) and $\ker(F)$ under $P^\top P$. We now present a general bound on the leakage:

Theorem 3.8. *Let $\{g'_j\}_{j=1}^k \subset \mathbb{R}^d$, and for each j let $g'_{j,\perp} := \Pi_{\ker(F)} g'_j$ denote the orthogonal*

projection of g'_j *onto* $\ker(F)$. *Let* $k' := \dim(g'_{j,\perp}\}_{j=1}^k))$. *For any* $\varepsilon, \delta \in (0,1)$, *if*

$$m = \Omega\left(\frac{r + \min\left\{\log(k/\delta), k' + \log(1/\delta)\right\}}{\varepsilon^2}\right),$$

then with probability at least $1-\delta$, *the following holds for all* $j \in \{1, \ldots, k\}$:

- ***Unregularized:*** $|\tilde{\tau}_0(g, g'_{j,\perp})| \le \varepsilon \|g\|_2 \|g'_{j,\perp}\|_2 / \lambda^+_{\min}(F)$, *where* $\lambda^+_{\min}(F)$ *denotes the smallest non-zero eigenvalue of* F.
- ***Regularized:*** $|\tilde{\tau}_\lambda(g, g'_{j,\perp})| \le \varepsilon \|g\|_2 \|g'_{j,\perp}\|_2 (\frac{1}{\lambda} + \frac{2\|F\|_2}{\lambda^2})$ *for any* $\lambda > 0$.

Proof sketch. The proof is organized around a deterministic reduction: Theorem 1.6 (in Appendix A.4) shows that both the unregularized and regularized leakage bounds follow as soon as two concentration conditions hold for the sketch P: (i) an operator-norm bound on $\mathrm{range}(F)$, $\|U^\top (P^\top P - I) U\|_2 \le \varepsilon$ for an orthonormal basis U of $\mathrm{range}(F)$, and (ii) a cross-term bound between $\mathrm{range}(F)$ and the kernel direction(s), $\|U^\top (P^\top P - I) g'\|_2 \le \varepsilon \|g'\|_2$. For a single test gradient $g'_\perp$, both conditions follow from applying Theorem 1.2 to the $(r+1)$-dimensional subspace $\mathrm{span}(\mathrm{range}(F) \cup \{g'\})$, which yields the claimed $m = \Omega((r + \log(1/\delta))/\varepsilon^2)$ scaling. To obtain uniform control over multiple test gradients, we use two complementary arguments: a *subspace argument*, which applies the same concentration bound to $\mathrm{span}(\mathrm{range}(F) \cup \{g'_{j,\perp}\}_{j=1}^k)$ and yields the dependence on $k' = \dim \mathrm{span}(\{g'_{j,\perp}\})$ (Theorem 1.8); or a *union-bound argument*, which establishes a fixed-g' tail bound and unions over k, yielding the $O(\log k)$ dependence (Theorem 1.9). □

3.2.2 Leakage of Factorized Influence

Theorem 3.8 is stated for oblivious sketches with i.i.d. rows. We now extend and prove an analogous leakage guarantee for factorized sketches $P = P_A \otimes P_E$ when F admits a Kronecker factorization.

Theorem 3.9. *Let* $A, E \succeq 0$ *and* $F := A \otimes E$, *with* $P = P_A \otimes P_E$ *be the same setting as Theorem 3.6, and let* $r_A := \mathrm{rank}(A)$, $r_E := \mathrm{rank}(E)$, *and* $r := \mathrm{rank}(F) = r_A r_E$. *Let* $\{g'_j\}_{j=1}^k$

be test gradients of the form $g'_j = a'_j \otimes e'_j$*, and write* $a'_j = a'_{j,\parallel} + a'_{j,\perp}$ *with* $a'_{j,\parallel} \in \mathrm{range}(A)$ *and* $a'_{j,\perp} \perp \mathrm{range}(A)$*, and similarly* $e'_j = e'_{j,\parallel} + e'_{j,\perp}$*. Define* $k_A := \sum_{j=1}^{k} \mathbb{1}(a'_{j,\perp} \neq 0)$*,* $k_E := \sum_{j=1}^{k} \mathbb{1}(e'_{j,\perp} \neq 0)$*, and* $k'_A := \dim(\mathrm{span}(\{a'_{j,\perp}\}_{j=1}^{k}))$*,* $k'_E := \dim(\mathrm{span}(\{e'_{j,\perp}\}_{j=1}^{k}))$*. For any* $\varepsilon, \delta \in (0,1)$*, if*

$$m_A = \Omega\left(\frac{r_A + \min\{\log(\frac{k_A}{\delta}), k'_A + \log(\frac{1}{\delta})\}}{\varepsilon^2}\right), m_E = \Omega\left(\frac{r_E + \min\{\log(\frac{k_E}{\delta}), k'_E + \log(\frac{1}{\delta})\}}{\varepsilon^2}\right),$$

then with probability at least $1-\delta$*, the following bounds hold simultaneously for all* $j \in \{1, \ldots, k\}$*:*

- ***Unregularized:*** $|\tilde{\tau}_0(g, g'_{j,\perp})| \leq \varepsilon \|g\|_2 \|g'_{j,\perp}\|_2 / \lambda^+_{\min}(F)$*.*
- ***Regularized:*** $|\tilde{\tau}_\lambda(g, g'_{j,\perp})| \leq \varepsilon \|g\|_2 \|g'_{j,\perp}\|_2 (\frac{1}{\lambda} + \frac{2\|F\|_2}{\lambda^2})$ *for any* $\lambda > 0$*,*

Proof sketch. The factorized theorem is proved by following the same high-level template as Theorem 3.8: we first reduce the leakage bound to the two concentration conditions in Theorem 1.6 (stability on $\mathrm{range}(F)$ and a cross-term bound between $\mathrm{range}(F)$ and $\ker(F)$). For a Kronecker sketch $P = P_A \otimes P_E$, the stability condition on $\mathrm{range}(F) = \mathrm{range}(A) \otimes \mathrm{range}(E)$ is obtained by controlling the factor-level subspace deviations $\|U_A^\top (P_A^\top P_A - I) U_A\|_2$ and $\|U_E^\top (P_E^\top P_E - I) U_E\|_2$ (with U_A, U_E bases of $\mathrm{range}(A), \mathrm{range}(E)$). For the cross-term condition, we expand $P^\top P - I$ into factor deviations and use Theorem 1.11 (in Appendix A.5) to reduce $\|U^\top (P^\top P - I) g'_\perp\|_2$ to a small collection of factor-level "primitive" quantities such as $\|U_A^\top (P_A^\top P_A - I)(\cdot)\|_2$ and $\|U_E^\top (P_E^\top P_E - I)(\cdot)\|_2$. Finally, as in the proof of Theorem 3.8, these primitives are controlled via a *union-bound argument* (yielding the $O(\log k_\cdot)$ terms) or a *subspace argument* (yielding the $k'_\cdot$ terms). Plugging these bounds into Theorem 1.6 yields the stated leakage guarantees; full details are in Appendix A.5. □

Remark 3.10. Which argument is tighter depends on the geometry of the test gradients. When $\{g'_j\}$ are strongly correlated or effectively low-dimensional, one can have $k' \ll k$, in which case the subspace argument is preferable. In contrast, in high ambient dimension,

moderately many generic test gradients are typically in general position, so k' rapidly grows to $\min\{k, d\}$ and in particular satisfies $k' \approx k$ once $k \ll d$. In this common regime, the union-bound argument yields the more practical scaling in k, requiring only an additional $O(\log k)$ sketch size to ensure uniform control.

3.3 Experiment and Discussion

We empirically illustrate several implications of our theory. Throughout, we consider F to be the empirical Fisher, and P to be the sparse JL transform [53], and we always report the results across 5 independent runs with different sampled P. Following the data attribution library `dattri` [68], we consider three dataset–model pairs: 1.) MNIST-10 + LR, 2.) MNIST-10 + MLP, and 3.) CIFAR-2 + ResNet9. Each setting uses 5000 training examples and 500 held-out test examples, so the empirical Fisher has rank at most $r \leq 5000$.

Firstly, we show the effective dimension $d_\lambda(F) = \sum_{i=1}^{r} \lambda_i/(\lambda_i + \lambda)$ can be much smaller than $r = \text{rank}(F)$. Specifically, Fig. 2 plots the ordered eigenvalues $\{\lambda_i\}_{i=1}^{r}$ of F. The spectrum decays quickly, hence for moderate λ, the terms $\lambda_i/(\lambda_i + \lambda)$ become small for large i, and consequently $d_\lambda(F)$ can be far smaller than r.

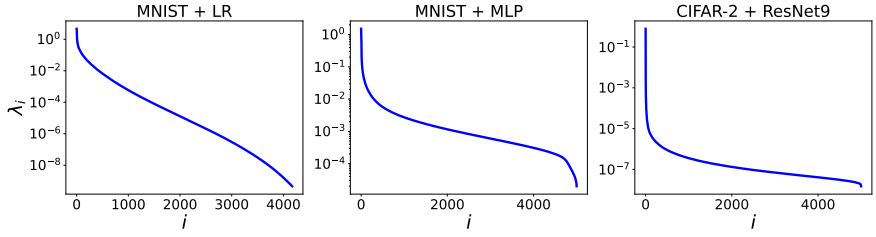


Figure 2: Ordered spectrum λ_i of the empirical Fisher F.

We next test the predictions of Theorems 3.2 and 3.8 by directly measuring the approximation error. Given $\lambda \geq 0$, we consider $\varepsilon_\lambda(g, g') = |\tilde{\tau}_\lambda(g, g') - \tau_\lambda(g, g')|/\sqrt{\tau_0(g, g)}\sqrt{\tau_0(g', g')}$ for gradients g and g', which is the normalized error considered in Theorem 3.2.

Fig. 3 supports the scaling predicted by our theory. Each curve plots the 95^{th} percentile of $\varepsilon_\lambda(g, g')$ against the normalized sketch size $m/d_\lambda(F)$. Once m is on the order of $d_\lambda(F)$,

the error begins to decay in the manner suggested by Theorem 3.2. Empirically, this indicates that (i) the hidden constant in the sketch-size requirement is modest and (ii) the additional leakage effect from Theorem 3.8 decreases quickly as m grows.

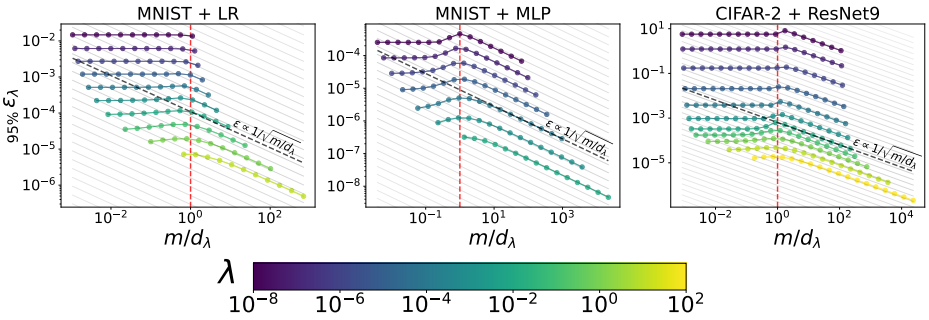


Figure 3: Approximation error versus normalized sketch size.

Faithfulness–Utility Tradeoff. A small approximation error does not necessarily imply strong downstream performance. In particular, optimizing ε_λ to be very small typically favors larger λ and larger sketch size m, because stronger regularization makes the influence computation less sensitive to sketching. As a result, the λ that minimizes ε_λ need not be the λ that maximizes downstream utility, especially when the curvature information in F is important for the task. We illustrate this using LDS [38], a standard metric in data attribution. Fig. 4 reports LDS over a range of sketch sizes and regularization strengths, and as we expect, the best-performing λ^* is typically intermediate.

These observations suggest a simple two-stage procedure. First, using a small validation set and a sufficiently large sketch size m, sweep over λ and select λ^* that maximizes the downstream metric. Second, fix $\lambda = \lambda^*$ and increase m until $m \gtrsim C d_{\lambda^*}(F)$, which ensures that the influence estimates are faithful. Fig. 5 illustrates this strategy for LDS: the square markers in the right panel (95^{th} percentile LDS) indicate how large m must be to approach the best attainable LDS. In our experiments, a constant $C \in (10, 100)$ is sufficient, making the dependence on $d_{\lambda^*}(F)$ operational.

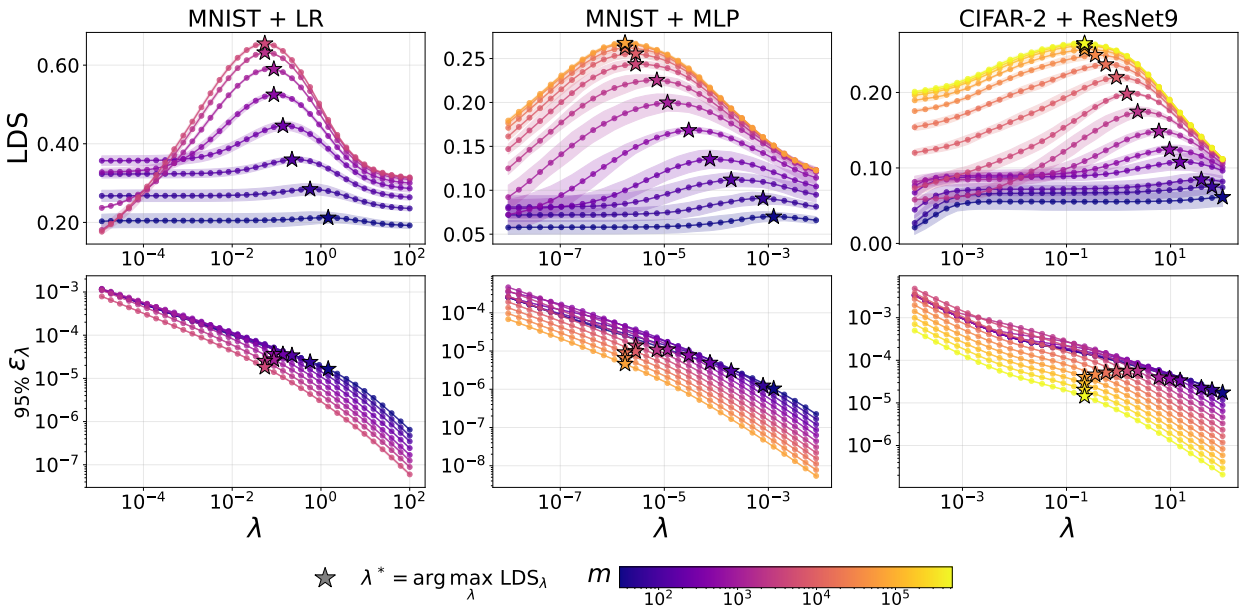


Figure 4: Approximation error and LDS versus λ.

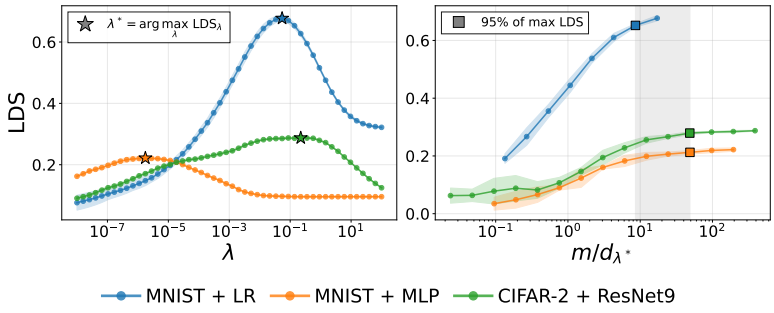


Figure 5: Left: selecting λ^* on a validation set using large m. Right: held-out test LDS versus $m/d_{\lambda^*}(F)$.

3.4 Related Work

Influence functions were originally introduced as a classical tool in robust statistics [35] and later adapted to machine learning by Koh and Liang [36]. Owing to their flexibility and generality, influence-based methods have since been widely applied to tasks such as data cleaning [69], model debugging [70], and subset selection [30], and have been extended to large-scale models, including large language models [62] and diffusion models [58]. However, applying influence functions to modern neural networks poses significant computational challenges due to the need to invert a high-dimensional, often rank-deficient, curvature matrix F [36, 71].

Several recent works propose scalable approximations based on random projection and related sketching techniques, where they typically project per-sample gradients into a lower-dimensional space before computing influence scores [37, 38, 71], sometimes in combination with explicit regularization [39, 50]. Despite their empirical success, the theoretical guarantees underlying these methods remain limited, and their correctness is often justified heuristically.

Specifically, existing theoretical justifications for projection-based influence methods typically appeal to the Johnson–Lindenstrauss (JL) lemma [51] in the data attribution literature [29, 37, 38]. Given a finite set of vectors of size n in $\mathbb{R}^d$, the JL lemma guarantees that $m = O(\log(n)/\varepsilon^2)$ suffices to approximately preserve the pairwise distances between the n points up to a $(1 \pm \varepsilon)$ factor. While powerful, this guarantee is fundamentally misaligned with the structure of influence functions. Influence scores are not determined by Euclidean distances between gradients, but by *inverse-sensitive* bilinear forms $\tau_0(g, g') = g^\top F^{-1} g'$ involving the inverse (or pseudoinverse) of a second-order matrix F, and sketching changes the operator to be inverted. Thus, preserving $\|Pg\|_2$ (even uniformly over a finite set) does not directly control either the stability of matrix inversion after projection, nor the resulting bilinear form.

3.5 Conclusion

In this work, we show that projection-based influence is governed by the interaction between the sketch and the curvature operator, and that conventional Johnson–Lindenstrauss arguments, which only control Euclidean geometry, are misaligned with inverse-sensitive influence computations [38, 50, 72]. By characterizing how projection interacts with common techniques such as ridge regularization and structured curvature approximations, our unified theory provides principled and actionable guidance for applying influence functions reliably at scale. More broadly, we view this work as a step toward a more principled understanding of scalable data attribution, and we hope it motivates further theoretical and empirical investigation into the interplay between projection, regularization, curvature, and evaluation criteria in influence functions.

Chapter 4 A Local Data Attribution Framework for Online Reinforcement Learning

While the previous chapter developed the theoretical foundations for scalable influence computation, this chapter turns to a new setting: online reinforcement learning. We ask whether data attribution can be extended beyond static supervised learning to settings in which the training data are generated by the model itself. In online RL, the agent continuously alternates between collecting experience and updating its policy, so each training sample can affect not only the current update but also the future data distribution induced by the evolving policy. This feedback loop breaks the assumptions underlying standard attribution methods designed for fixed datasets and fixed objectives. To address this challenge, this chapter develops a local attribution framework for online RL, based on TracIn-style gradient alignment, that traces an agent's behavior back to individual training experiences and enables both interpretability and practical improvements to training.

Reinforcement learning (RL) has achieved remarkable success across a wide range of sequential decision-making problems, including game playing [73, 74], robotic control [75], and the alignment of large language models [12]. Among its variants, online RL methods such as A3C [76] and PPO [77] are especially appealing in real-time, adaptive, and safety-critical settings because they interleave data collection with policy optimization, allowing agents to respond to changing environments on the fly [75, 78]. Yet this same closed-loop nature also makes online RL difficult to understand and control: training is often sample-inefficient, high-variance, and unstable, frequently requiring millions of interactions and exhibiting substantial variability across runs [79, 80, 81].

These challenges motivate a finer-grained understanding of how individual training experiences shape learned behavior. Prior work on RL interpretability has explored a range of approaches [82, 83], but many of them do not support example-level explanations or are not readily actionable for improving training (see Sec. 4.5). Data attribution offers a complementary perspective by tracing model behavior back to the data that produced it,

and has proved useful in other machine learning settings for tasks such as data selection [84], bias mitigation [85], and fact tracing [86]. However, extending data attribution to online RL is fundamentally nontrivial because the training data are endogenous: each experience both updates the policy and influences the future experiences that the policy will collect.

In this chapter, we address this gap by developing, to our knowledge, the first data attribution framework for online RL, with a particular focus on the widely used Proximal Policy Optimization (PPO) algorithm [77]. Our contributions are threefold:

1. **A principled and flexible framework (Sec. 4.2).** We propose a local data attribution framework for online RL, interpreting model checkpoints w.r.t. the records from the recent training buffer. We define the attribution entity as the atomic unit in PPO training, design two target functions that capture agent actions and cumulative returns, and measure each record's influence through gradient similarity between its training loss and the target.

2. **Fresh insights into learning (Sec. 4.3).** We demonstrate the power of our framework through three applications: a) *diagnosis of learning*: we show records most harmful for learning feature inaccurate advantage estimates; b) *temporal analysis of behavior formation*: we reveal an intriguing phase transition of critical records in shaping agent behaviors; c) *targeted intervention*: we show that removing records with the most negative influences can effectively improve model training.

3. **Improved training (Sec. 4.4).** Building on the targeted intervention, we further develop an iterative influence-based filtering algorithm (IIF) that significantly improves standard online RL training. Across standard RL benchmarks to modern RLHF for large language models, IIF consistently improves *sample efficiency*, reduces *computational cost*, and enhances *final performance*.

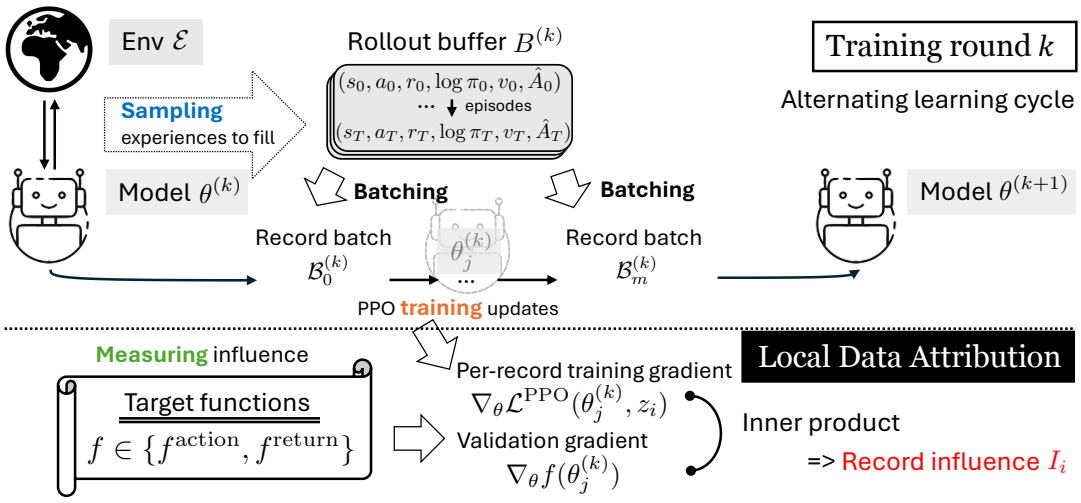


Figure 6: An Illustration of the alternating learning cycle in online RL (Sec. 4.1.1) and our local data attribution framework (Sec. 4.2.1). Online RL operates in alternating rounds of data collection and policy updates; our local data attribution framework quantifies how *individual* records from a single round influence policy update in that round.

4.1 Preliminaries

4.1.1 Online Reinforcement Learning

We consider the online RL setting, where an agent learns to maximize long-term returns by interacting with the environment. The environment $\mathcal{E}$ is modeled as a Markov Decision Process (MDP) defined by the tuple $(\mathcal{S}, \mathcal{A}, P, R, \gamma, d_0)$, where S is the state space, $\mathcal{A}$ the action space, P the transition function, R the reward function, $\gamma \in [0, 1]$ the discount factor, and $d_0 \in \mathcal{P}(\mathcal{S})$ the initial state distribution. At timestep t, the agent observes s_t, takes action a_t, receives reward r_t, and transitions to s_{t+1}.

Online RL typically proceeds in alternating **training rounds** of data collection and model training (Fig. 6). In round k, the data collection phase involves the agent executing the current policy $\pi_{\theta^{(k)}}$, sampling experiences over multiple episodes to accumulate n transition records in a rollout buffer $B^{(k)}$. Each record contains the raw *transition* (s_t, a_t, r_t) and several computed quantities, including the action log probability $\log \pi_{\theta^{(k)}}(a_t|s_t)$, estimated value v_t, and advantage estimate $\hat{A}_t$. Model parameters are then updated iteratively starting from $\theta_0^{(k)} = \theta^{(k)}$: at optimization step j, training on the mini-batch $\mathcal{B}_j^{(k)}$ drawn

from $B^{(k)}$ updates parameters from $\theta_j^{(k)}$ to $\theta_{j+1}^{(k)}$. In this chapter, we focus on Proximal Policy Optimization (PPO), a widely used, effective algorithm in various applications [12, 75, 87].

Proximal policy optimization (PPO) [77]. PPO is a policy gradient method for online RL that optimizes a clipped surrogate function. The core PPO objective, which is typically combined with a value function loss and an entropy bonus during optimization, is defined as:

$$\mathcal{L}^{\text{PPO}}(\theta) = \mathbb{E}_{(s,a)\sim\mathcal{B}_j^{(k)}}\left[\min\left(\frac{\pi_\theta(a|s)}{\pi_{\theta^{(k)}}(a|s)}\hat{A}(s,a), \text{clip}\left(\frac{\pi_\theta(a|s)}{\pi_{\theta^{(k)}}(a|s)}, 1-\epsilon, 1+\epsilon\right)\hat{A}(s,a)\right)\right],$$

where ϵ is a hyperparameter that limits policy changes between rounds and promotes stable learning.

4.1.2 Data Attribution via TracIn

Building on TracIn introduced in Sec. 2.1, this chapter adopts it as our primary attribution tool due to its conceptual simplicity, relative efficiency, and widespread use in recent works [84, 88, 89]. We briefly recall that the TracIn score of a training sample z_i with respect to a target function $f(\theta)$ is $\text{TracIn}(z_i) = \sum_{j:\, z_i\in\mathcal{B}_j} \eta_j \nabla_\theta f(\theta_j)\cdot\nabla_\theta \ell(\theta_j, z_i)$, measuring the cumulative gradient alignment between training and target across optimization steps.

4.2 Framework Design

Online RL presents unique challenges for data attribution, due to the way data interacts with model parameters during learning. To tackle this challenge, we introduce a *local* attribution framework tailored to *local* policy optimization inherent in online RL.

Challenges. The key feature of online RL is *the circular dependency between data and model*—earlier experiences drive policy updates, and updated policies produce new experiences to learn from. The dependency of data on model (red arrows in Fig. 7) is

unique to online RL and cannot be addressed by existing attribution methods. Current data attribution methods include *retraining-based* (e.g., Ghorbani and Zou [90]) and *gradient-based*, with the latter further divided into *static* and *dynamic* [34]. Retraining-based methods require training the model once for each of the records being evaluated, which is computationally expensive in any setting and particularly prohibitive in RL. Static methods implicitly assume model parameters are obtained from solving an empirical risk minimization problem over a fixed dataset, which is violated in the non-stationary, sequential data setting here. While dynamic methods (e.g., TracIn) capture the temporal dependencies of training data influences on model parameters, they still fail to account for this key effect of *data-model dependency*. If we compute influence scores using the original formulas from standard supervised learning, they capture only the impact on parameter updates, ignoring the extra *channel* of influences through future data generation. As a result, the scores may deviate significantly from the true influence we seek to measure. Furthermore, quantifying influences through this channel is challenging because sampling is stochastic and non-differentiable.

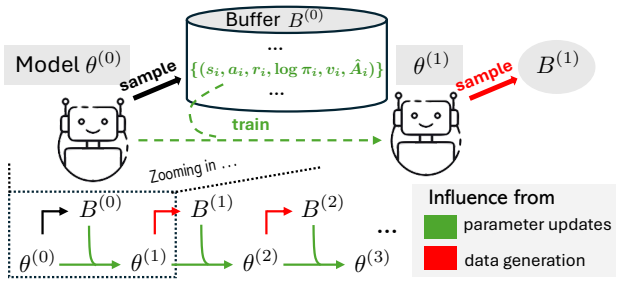


Figure 7: Twofold data influence: driving policy updates, shaping future data collection.

4.2.1 A Framework of Local Data Attribution

Our local data attribution framework addresses the circular data-model dependency. Online RL involves a *local policy optimization* structure, i.e., round k optimizes on a fixed buffer $B^{(k)}$ of on-policy data. Thus, each round serves as a natural unit of analysis. Our framework operates at this level, examining how records in $B^{(k)}$ contribute to the updates from $\theta^{(k)}$ to

$\theta^{(k+1)}$. This circumvents the challenges in tracing influence through the complex, cascading, and non-differentiable dependencies across the training history. Below, we detail the three key components of our framework.

Entity of attribution. We consider attribution to individual training records in the rollout buffer, $z_i = (s_i, a_i, r_i, \log \pi_i, v_i, \hat{A}_i)$, collected from the environment using the current policy $\theta^{(k)}$. These records form the *atomic* unit used in PPO updates and provide a natural granularity for attribution.

Target functions. Training data influence is usually reflected through the impact on model behaviors. Here we focus on two core aspects of an RL agent: agent action and cumulative return.

Agent action: To identify records influencing the agent's decision to take a specific action a at state s, we define a straightforward target function:

$$f^{\text{action}}(\theta) := \log \pi_\theta(a \mid s).$$

Cumulative return: We aim to understand which experience records contribute positively or negatively to the agent's ability to maximize cumulative return. Formally, the ideal quantity is the expected return $J(\theta) = \mathbb{E}_{\tau \sim \pi_\theta}[R(\tau)]$, where $R(\tau) = \sum_{t=0}^{T-1} r_t$ and trajectories τ are sampled by executing π_θ. However, using $J(\theta)$ directly poses two fundamental challenges. *First*, unlike supervised learning with a fixed validation set, the data distribution in online RL is inherently policy-dependent. This intertwining of policy and evaluation means no fixed, universal validation set exists. *Second*, raw returns $R(\tau)$ exhibit high variance, leading to noisy influence estimates.

To address these challenges, we introduce a stable surrogate objective based on a reference policy π^{ref} and advantage estimates $\hat{A}^{\text{ref}}$:

$$f^{\text{return}}(\theta) := \mathbb{E}_{\tau \sim \pi^{\text{ref}}, (s,a) \sim \tau} \left[\log \pi_\theta(a \mid s) \hat{A}^{\text{ref}}(s, a) \right].$$

This target function is structurally equivalent to the objective of REINFORCE with a baseline [91, Section 13.4]. By sampling from π^{ref}, we obtain a fixed evaluation distribution; using advantage estimates significantly reduces variance compared to raw returns. Maximizing $f^{\text{return}}(\theta)$ encourages increasing the probability of better-than-average actions and decreasing worse-than-average ones, capturing the essence of improving expected return while being tractable.

For attribution in round k, we set the reference policy $\pi^{\text{ref}} = \pi_{\theta^{(k)}}$, i.e., the policy snapshot at the beginning of the round. This is a key design choice of our *contextual* framework, which enables us to ask: *For the agent at its current stage of training, which experiences will be most helpful or harmful for the next update?* Unlike a fixed, off-distribution reference that may provide misleading signals due to mismatch with the agent's current state, our dynamic reference evolves with training, providing a stable and relevant basis for meaningful evaluation and attribution. Furthermore, since the training rollout buffer $B^{(k)}$ is collected under $\pi_{\theta^{(k)}}$, we can directly use it as the validation dataset. We provide further discussions on this design choice in Sec. 4.3.3 and Sec. 4.4.1.

We note that one key contribution in our framework is the design of *tractable yet meaningful* target functions, particularly f^{return}, which can be reused in future work with alternative attribution methods.

Remark 4.1 (Use cases of the two target functions). The two target functions have different use cases. f^{action} is mainly for *diagnosis*: understanding why the agent takes a specific action at a specific state (Sec. 4.3.2). On the other hand, f^{return} assesses contribution to overall performance, which makes it suitable for both *analysis* (Sec. 4.3.1) and *algorithmic policy improvement* (Sec. 4.4).

Method of attribution. We adapt TracIn to our online RL setting. For record z_i in the rollout buffer $B^{(k)}$, we compute its *influence score* by summing over the optimization

steps j within round k:

$$I_i := \sum_{j: z_i \in \mathcal{B}_j^{(k)}} \left\langle \nabla_\theta f(\theta_j^{(k)}), \nabla_\theta \mathcal{L}^{\text{PPO}}(\theta_j^{(k)}, z_i) \right\rangle, \quad \text{where } f \in \{f^{\text{action}}, f^{\text{return}}\}.$$

Here, $\nabla_\theta f(\theta_j^{(k)})$ is the gradient of the target function evaluated at $\theta_j^{(k)}$, and $\nabla_\theta \mathcal{L}^{\text{PPO}}(\theta_j^{(k)}, z_i)$ is the per-sample gradient of the PPO training objective for record z_i. We also discuss two design choices in Sec. 4.4.1 which substantially reduce the computational and storage costs of the vanilla TracIn.

Finally, we clarify how to interpret the computed influence scores. Records with positive influence *benefit* behavior formation or learning, whereas those with negative influence *harm* it. We refer to records with the most positive influence as *top records* and those with the most negative influence as *bottom records*; these terms will be used throughout the remainder of the chapter.

Remark 4.2 (Extension to other online RL algorithms). While we focus on PPO in our study, our framework readily extends to other online RL algorithms. For on-policy methods[4] such as TRPO [93] and A3C [76], the adaptation only requires modifying the per-sample loss gradient. For offline methods like DQN [94], we need to additionally change the target function to the Bellman error. In all cases, our attribution framework reveals whether training records help or hinder learning at the agent's current state. A key distinction is that on-policy methods allow direct validation with current data, whereas off-policy methods require sampling fresh rollouts.

4.3 Applications of Local Data Attribution

We now illustrate the practical value of our framework. The framework delivers fresh insights for RL researchers and practitioners, enabling key applications such as diagnosis of learning, temporal analysis of agent behavior formation, and targeted interventions

[4]For GRPO [92], which uses a group-relative baseline rather than value-function baseline, the target function needs to be adjusted as well.

during training. We demonstrate these capabilities through extensive empirical studies spanning a range of RL environments and tasks.

Experimental setup. We perform evaluation on a diverse suite of RL environments—navigation (`FrozenLake` and `MiniGrid`), classic control (`Acrobot` and `LunarLander`), driving (`Highway`), and locomotion (`BipedalWalker`)—covering discrete and continuous state and action spaces with varying complexity and reward structures. We defer descriptions of environments to Appendix B.1.1 and PPO training setups to Appendix B.1.2.

4.3.1 Diagnosis of Learning: What Features Bottom Records?

In this section, we analyze the bottleneck that hinders learning in online RL. Specifically, we examine the bottom records for f^{return} and uncover a consistent pattern across training rounds (additional examples in Appendix B.2.1): these bottom records are characterized by *inaccurate advantage estimates*, echoing observations in the literature [95].

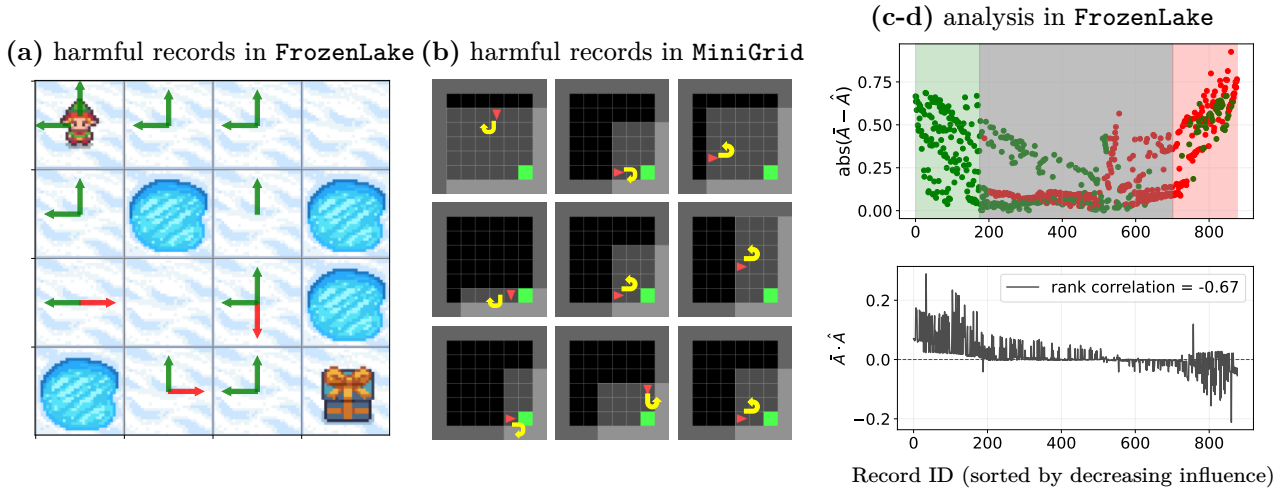


Figure 8: **(a-b) Examples of bottom records**. (a) Bottom 100 records in `FrozenLake` at $k = 5$, aggregated over (s, a) for demonstration: arrow indicates action, green/red for positive/negative $\hat{A}$. (b) Selected records among bottom 20 in `MiniGrid` at $k = 5$: ▼–agent, ▪–goal, gray area–the limited egocentric observation, yellow arrows–agent action in $\{\text{turn left}, \text{turn right}, \text{forward}\}$; all records shown are of positive $\hat{A}$. **(c-d) These records are harmful due to their inaccurate advantage estimates**. We sort records by decreasing influence (top on the left). (c) y axis is $|\bar{A} - \hat{A}|$; points with same/opposite signs for $\hat{A}$ and $\bar{A}$ colored green/red; top/bottom 20% region shaded green/red, and the intermediate in gray. (d) The product $\bar{A} \cdot \hat{A}$ versus record rank, showing a strong negative correlation.

Fig. 8(a–b) illustrates two examples. In `FrozenLake`, bottom records include poor actions receiving high positive $\hat{A}$ and good actions receiving negative $\hat{A}$. Similarly, in `MiniGrid`, the agent drifts from the goal but receives positive $\hat{A}$. These instances of *misleading* advantage estimates harm the learning.

We conduct quantitative analysis to characterize what constitutes "inaccurate" advantage estimates. We approximate the true advantage $A^\pi(s, a)$ using Monte Carlo (MC) rollouts from each (s, a), averaging over multiple trajectories (details in Appendix B.2.4). We refer to this as the MC estimate, denoted by $\bar{A}$, and compare it with the advantage estimate $\hat{A}$. We perform analysis in `FrozenLake`.

Our analysis reveals two key aspects of "inaccuracy": (1) **Sign mismatch**: A significant proportion of bottom records exhibit opposite signs for the advantage estimate $\hat{A}$ and the MC estimate $\bar{A}$ (marked by red points in Fig. 8(c)). (2) **Large magnitude errors**: These records also have large $|\bar{A} - \hat{A}|$. Together, sign flips and large magnitude errors generate strong but misleading learning signals. Indeed, the Spearman rank correlation [96] between each record's influence and the product $\bar{A} \cdot \hat{A}$ is strongly negative (Fig. 8(d)), confirming that misaligned advantages drive harmful gradient steps.

4.3.2 Temporal Analysis of Behavior Formation: Phase Transition of Top Records

We investigate the reinforcement of a specific behavior (a at s), characterized by a monotonic increase in $\pi(a|s)$. We track the evolution of top records w.r.t. f^{action} across training rounds, which are critical in shaping the agent's behavior. Our analysis reveals an intriguing three-stage phase transition (Fig. 9).

1. **Initial association**: Initially, top records highlight patterns based on simple *action-advantage association*: they manifest target action paired with positive $\hat{A}$, or alternative actions paired with negative $\hat{A}$ (see Appendix B.2.2 for examples). The agent's behavior in this phase is reinforced through this naive association, largely ignoring the context

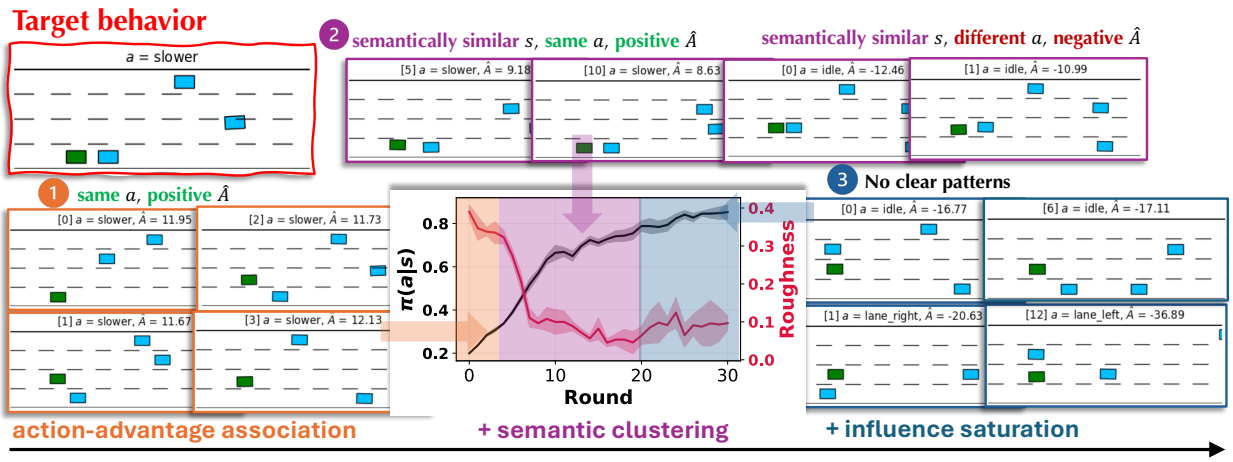


Figure 9: **Phase change of top records** in `Highway`, with the target behavior *taking the action "slower" when tailing the front vehicle*. In the inner plot, the black curve depicts $\pi(a|s)$; the red curve shows the measured roughness of the graph. ▬: ego vehicle; ▬: other vehicle. Three phases: ❶: simple action-advantage associations; ❷: semantic clustering (tailing states); ❸: no clear patterns.

of *state*. This basic association persists throughout training, even as more complex relationships are learned.

2. **Semantic clustering**: As learning progresses, the agent develops more nuanced representations. As a result, a pattern of *semantic clustering* develops alongside the initial action-advantage association. Top records in this phase demonstrate action-advantage association *within* states semantically similar to the target state, indicating the agent has learned to generalize across similar situations.

3. **Influence saturation**: In the final phase where learning approaches convergence, influence scores for most records stabilize near zero and become dominated by noise. Due to this noise, the top records appear less structured, though the action-advantage association still persists.

We quantify these phases by analyzing the *roughness* (normalized Dirichlet energy) [97] of a similarity graph, a measure closely related to the graph Laplacian [98]. In this graph, nodes represent records, values are (L_∞-normalized) influence scores $\tilde{I}_i$, edge weights w_{ij} capture semantic similarity and decay with embedding distance (details in Appendix B.2.2). Roughness, computed as $\sum w_{ij}(\tilde{I}_i - \tilde{I}_j)^2 / \sum w_{ij}$, is low when semantically similar records have

similar influence; this captures the *clustering* effect. We track roughness across training rounds. As Fig. 9 shows, roughness remains high in Phase 1, indicating influence scores are largely uncorrelated with semantic similarity. It then significantly drops in Phase 2, representing the formation of semantically meaningful *clusters* of records with similar influences. In Phase 3, roughness remains low due to the settling of clustering, but exhibits minor fluctuations due to influence scores dominated by noise upon convergence.

4.3.3 Targeted Interventions During Training: Filtering Amplifies Policy Gain

Sec. 4.3.1 demonstrates that our framework can identify harmful training records, thereby opening possibilities for targeted interventions. As a sanity check, we apply a simple intervention procedure within *a single training round* to verify if removing these records yields performance gains.

Our procedure is straightforward: in round k, we identify records in $B^{(k)}$ with negative influence scores w.r.t. f^{return}, remove them, and re-train the agent on the filtered dataset starting from $\theta^{(k)}$. Fig. 10 shows that this consistently improves performance throughout learning and across environments.

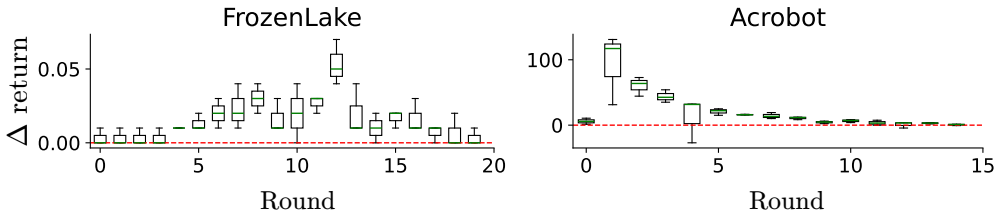


Figure 10: **Boxplots of Δ return for single round interventions in two environments**; red dashed line for zero Δ. We intervene for each round *independently*. The Δ return is computed as the difference between the test return of the model trained on the *filtered* dataset and the *original* dataset. Results are shown for 3 random seeds. Additional results can be found in Appendix B.2.3.

A reader may ask: how can f^{return} be meaningful when it relies on on-policy data with potentially inaccurate advantage estimates, unlike clean validation data used in traditional data attribution for supervised learning? Despite potential noise in individual records, the aggregated signal from f^{return} is reasonably robust. This arises from the close alignment

of f^{return} with the PPO objective: effective PPO updates on the training buffer imply a reliable f^{return} for attribution, enabling our intervention to clear away misleading records while retaining beneficial ones. This can be seen as *purifying* the learning signal, thereby *amplifying* the improvement achieved by PPO. More discussions are in Appendix B.2.3.

4.4 Iterative Influence-Based Filtering for Online RL Training

Standard online RL algorithms typically treat all collected experiences uniformly. However, as our analysis in Sec. 4.3.1 has shown, some records can be harmful for learning. This likely contributes to the notorious *sample inefficiency* of online RL, a challenge widely acknowledged [80]. Given this, a natural question arises: *can we leverage the local data attribution framework to tackle this challenge?*

We propose Iterative Influence-Based Filtering (IIF), building on the single-round interventions in Sec. 4.3.3. IIF filters records based on their computed influence scores, uses the resulting improved policy to sample new data, and repeats the cycle. This creates a loop for iterative refinement. We detail the algorithm below and showcase its effectiveness in traditional RL environments and RLHF for LLMs.

4.4.1 Algorithm and Designs

Algorithm 1: Iterative Influence-Based Filtering (IIF) for Online RL

Define: $\mathcal{E}$: environment. n: # records in a rollout buffer. $p \in (0, 1]$: percentage of negative records to drop.

Function `Update(model)`:
  ▷ `Stage I: sampling`
  $B \leftarrow$ `CollectTransitions(`$\mathcal{E}$`, model, `n`)` ▷ `collect transitions into buffer` B
  ▷ `Stage II: Filtering`
  $I \leftarrow$ `ComputeInfluence(model,` B`)` ▷ `compute influence for each record`
  $B_{\text{filtered}} \leftarrow$ `DiscardBottomRecords(`B`,` I`,` p`)` ▷ `drop bottom records`
  ▷ `Stage III: training`
  return `PPOUpdate(model,` B_{filtered}`)`
for iter $= 1$ **to** T **do**
  model $\leftarrow$ `Update(model)`

Alg. 1 outlines IIF. Compared to standard PPO, IIF introduces an additional step of

filtering (in red) between data collection and training. We further highlight the desiderata and IIF's design choices.

Sample efficiency. We aim to reduce the environment interactions required to reach a given performance level. To achieve this, IIF reuses the original rollout buffer $B^{(k)}$ as the validation set for influence calculation, incurring no extra sampling overhead. Furthermore, by selectively filtering bottom records, IIF accelerates learning, thus further reducing the total interactions needed.

Computational cost. We aim to keep the overhead of influence calculation small. This is achieved through two design choices. (1) Instead of iterating over all intermediate checkpoints, we compute the influence scores for the entire rollout buffer $B^{(k)}$ in round k via $\left\langle \nabla_\theta f(\theta^{(k)}), \nabla_\theta \mathcal{L}^{\text{PPO}}(\theta^{(k)}, z_i) \right\rangle$, using only the initial parameter $\theta^{(k)}$. This saves a full training pass and excessive forward/backward calculations. (2) We implement an efficient "ghost dot product" following Wang et al. [99].

Final performance. We aim to improve the policy's final performance compared to standard training. IIF fulfills this through identifying and filtering out harmful records.

IIF employs a hyperparameter, p, which determines the amount of records to discard. We evaluate various p's and report the best in Fig. 11. We observe that removing all negative-influence records ($p = 100\%$) as in Wang et al. [99] is often suboptimal, likely due to the non-additivity of sample influence [30]. Full ablation and recommendations for the choice of p are in Appendix B.2.6.

4.4.2 Experiments in Traditional RL Environments

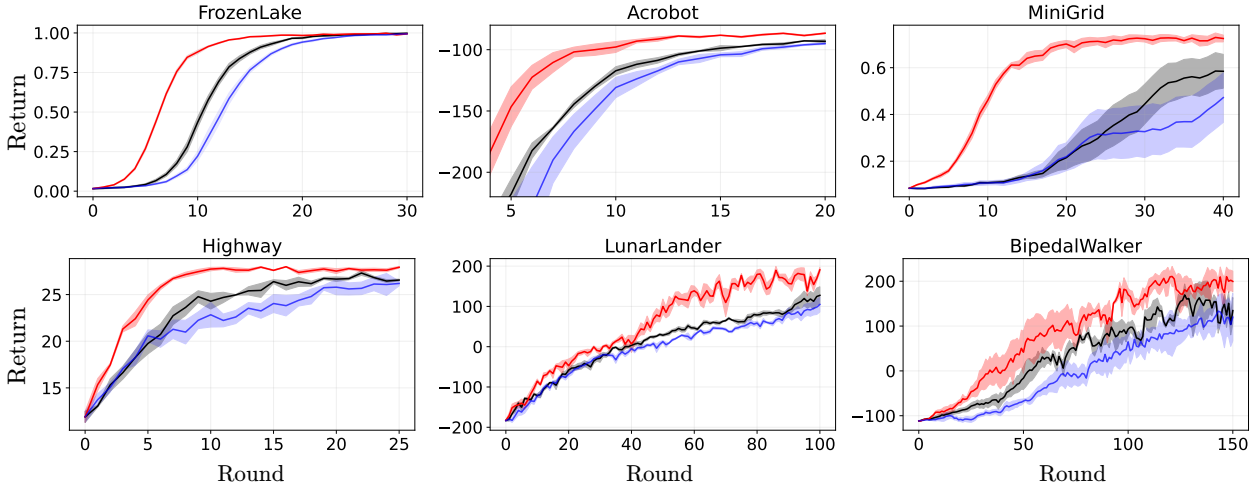
Experimental setup. We evaluate IIF on the diverse set of RL environments introduced in Sec. 4.3.

Baselines: We compare IIF with standard PPO and a random filtering baseline (dropping a similar fraction of records). We additionally investigate an advantage based filtering heuristic in Appendix B.2.4 motivated by the characterization of bottom records in Sec. 4.3.1,

as well as a TD error based heuristic in Appendix B.2.5 inspired by the Prioritized Experience Replay algorithm [100].

Metrics: We quantify sample efficiency by the reduction in training rounds required for IIF to match standard training. For a performance level v (measured by test return), let $m_{\text{std}}(v)$ and $m_{\text{IIF}}(v)$ be the earliest training rounds where standard training and IIF achieve performance at least v, respectively. The reduction at v is defined as $(1 - {}^{m_{\text{IIF}}(v)}/_{m_{\text{std}}(v)}) \times 100\%$. We report two metrics: SE_{ave}, the mean reduction over a list of strictly increasing performance levels reached by standard training, and SE_{peak}, the reduction at its peak. We measure computational cost by runtime; we similarly define RT_{peak} as the reduction of runtime at the performance peak. Model performance is measured by the average test return over multiple episodes. See Appendix B.1.2 for further experimental setups.

(a) Test returns over training rounds. IIF (Ours) Standard Random



(b) Improvement in sample efficiency and runtime

	`FrozenLake`	`Acrobot`	`MiniGrid`	`Highway`	`LunarLander`	`BipedalWalker`
SE_{ave} (↑)	34.0% ± 2.0%	36.7% ± 6.5%	65.8% ± 3.3%	37.7% ± 6.1%	26.0% ± 1.8%	31.0% ± 8.7%
SE_{peak} (↑)	19.2% ± 5.9%	48.5% ± 0.8%	61.7% ± 4.1%	55.1% ± 2.9%	39.7% ± 3.7%	26.2% ± 8.0%
RT_{peak} (↑)	29.5% ± 2.9%	55.2% ± 1.0%	69.1% ± 1.7%	59.9% ± 0.7%	44.9% ± 2.5%	29.2% ± 0.7%

Figure 11: (a) **Test returns over rounds for IIF vs. baselines.** IIF speeds up learning and improves performance. Results are averaged over 5 random seeds. For `Acrobot`, we omit early rounds where returns rise from -500 to -200 for better visualization. (b) **Sample efficiency and runtime metrics.**

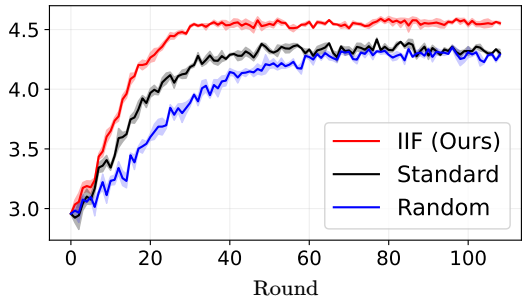
Results. Fig. 11(a) presents the test returns for each environment; Fig. 11(b)

summarizes the efficiency and runtime metrics. We report a detailed breakdown of runtime in Appendix B.2.9. Our key findings are summarized as follows: 1) IIF achieves substantial sample efficiency gains, showing a 20-67% reduction in training rounds required to match the standard training performance across environments. 2) The computational overhead of IIF is negligible, and offset by the reduced optimization time (see Appendix B.2.9), leading to significant improvement in runtime. 3) IIF's final performance exceeds standard training in almost every environment. These observed gains stem from effective data attribution rather than mere data reduction: random filtering performs significantly worse than original training.

4.4.3 Extending IIF to RLHF for Large Language Models

As the final part, we apply IIF to improve Reinforcement Learning from Human Feedback (RLHF).[5] Compared to standard PPO, RLHF introduces several key differences. First, the atomic unit shifts from state-action records to prompt-generation pairs, where each generation is a *trajectory* (or sequence) of tokens. Second, RLHF incorporates *dual* reward sources: a reward model evaluating the final generation, and a per-token KL divergence penalty to constrain deviation from a reference model.

(a) Training Reward (↑)



(b) Test toxicity (↓) on a different test set, evaluated using a different toxicity detector.

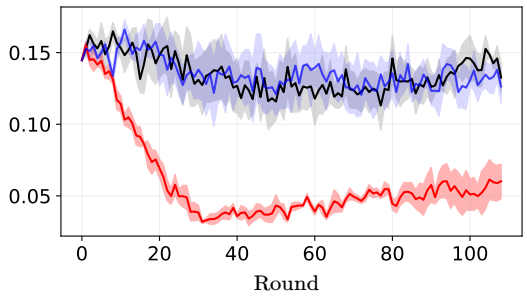


Figure 12: IIF improves the efficiency and performance of RLHF.

To accommodate these differences, we adapt IIF for RLHF by employing a sequence-level

[5]Another line of work focuses on improving reward modeling in RLHF (the stage before PPO) via preference data selection [101, 102, 103]; this is orthogonal to our work.

objective:

$$f^{\text{seq}}(\theta) = \mathbb{E}_{x\sim D_{\text{val}}, y\sim\pi^{\text{ref}}(\cdot|x)}\left[\log \pi_\theta(y \mid x)\hat{A}^{\text{ref}}_{-1}(x, y)\right],$$

where x is a prompt drawn from the validation set D_{val}, y the generation, $\log \pi_\theta(y|x) = \sum_i \log \pi_\theta(y_i|x, y_0, \ldots, y_{i-1})$ the log-probability of the sequence y given x, and $\hat{A}^{\text{ref}}_{-1}$ the advantage estimate at the last token. This objective emphasizes the reward model's feedback at the last token.

Experimental results: toxicity mitigation. We consider the task of detoxifying LLMs using RLHF [104], using gpt-neo-2.7B [105] as our base model. Fig. 12 illustrates the effectiveness of our approach. We defer detailed experimental setups to Appendix B.1.3 and additional results (e.g., comparisons with using the target function f^{return}) in Appendix B.2.11.

We further highlight IIF's substantial gains in *computational efficiency*. IIF filters out negative-influence records ($\sim$50% of all), effectively *halving* the optimization time per round. Furthermore, IIF accelerates learning, requiring less than *half* the number of rounds to surpass standard training, significantly enhancing sample efficiency. The overhead of influence calculation is minimal. Collectively, these factors result in an $\sim 4\times$ reduction in total runtime (detailed breakdown in Appendix B.2.12).

4.5 Related Work

Interpretability in reinforcement learning has become a central research theme because real-world deployment requires agents that are trustworthy and reliable [82, 83, 91, 106]. Early studies emphasize *feature*-level explanations: they highlight regions of the observation space that most influence an agent's decisions, often through saliency maps or attention heatmaps [107, 108, 109, 110, 111]. A complementary thread seeks *policy*-level explanations. These works approximate learned policies with human-interpretable rules [112, 113], design transparent architectures [114, 115], or dissect reward functions to clarify action choices [116, 117]. More recently, researchers have probed how entire training *trajectories*

shape behavior [118].

Zooming in further, identifying critical *states* offers a finer-grained view of decision making. Several approaches address offline settings [119, 120, 121, 122]. Closer to our focus are methods that target online RL such as lazy-MDP [123], StateMask [124] and RICE [125]. Lazy-MDP augments the action space with a "lazy" action and penalizes non-lazy choices; states where the agent still acts are interpreted as important. However, this approach requires modifying the training pipeline. StateMask and RICE train an auxiliary mask network alongside the policy, forcing random actions in selected states while keeping returns roughly unchanged; masked states are deemed non-critical. Nevertheless, these methods crucially rely on the policy being sufficiently developed, which limits their applicability when agents are still learning in complex environments.

Moving beyond these constraints, our work introduces data attribution as a principled lens for interpretability in online RL. This approach closes a key methodological gap in the literature, delivers fresh insights for RL researchers and practitioners, and informs more efficient and effective training.

4.6 Conclusion

This work pioneers data attribution for online RL by introducing a local attribution framework that addresses the circular dependency between data and model. The framework provides fine-grained insights into how training records shape model behaviors and offers a principled approach to enhancing the interpretability, efficiency, and effectiveness of online RL.

Chapter 5 Empirical Privacy Variance

The preceding chapters examined how training data shape model behavior. This chapter turns to a complementary facet of the same overarching question: when generative models are trained on sensitive data, how can we characterize and protect the privacy of the information they encode? Although differential privacy has become the dominant formal framework for privacy protection, its practical implications for generative models remain less well understood. This chapter critically examines the relationship between formal privacy guarantees and the privacy risks that models exhibit in practice. In the setting of language model fine-tuning, we show that even under the same nominal DP guarantee, different hyperparameter choices in DP-SGD can induce substantially different empirical privacy outcomes.

Modern large language models (LLMs) demonstrate remarkable proficiency on a wide range of tasks, from traditional ones such as summarization, to complex problem solving that involves reasoning and coding [126, 127, 128]; these capabilities arise from large-scale pre-training on massive datasets [129, 130, 131]. To enhance domain specialization and personalization, a common practice is to fine-tune pre-trained models on downstream tasks with user-contributed data [132, 133]. However, this process introduces significant privacy concerns, as datasets often contain sensitive information of individuals or organizations, which could be memorized and potentially divulged by LLMs [17, 134, 135, 136, 137].

In response, differential privacy (DP; [41]), a widely-adopted standard for privacy protection, has been incorporated into various stages of the LLM training pipeline, leading to a fruitful line of research advancing the utility-privacy trade-off in LLMs [46, 48, 138, 139, 140]. Nevertheless, privacy in LLMs is a nuanced concept, stemming not only from the unstructured and context-dependent nature of private information in natural language [141], but also from the generative nature of these models: during real-time user-model interactions, LLMs can inadvertently regurgitate private information, and such leaks are immediately made apparent to users [142, 143]. This reflects a pragmatic view on privacy centering

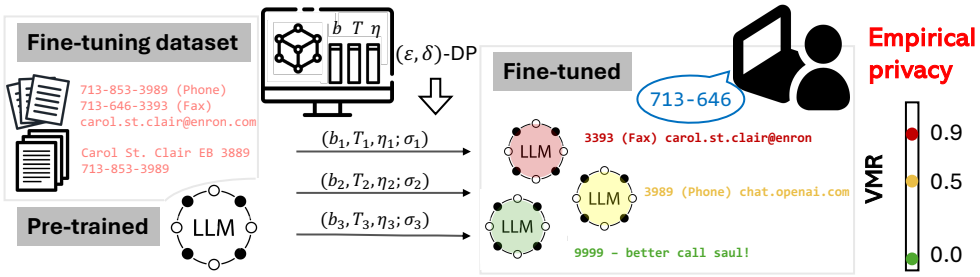


Figure 13: **Empirical privacy variance**: Starting from the *same* pre-trained model and fine-tuning on the *same* dataset (to achieve decent utility), DP-SGD with *different* hyperparameter configurations—each calibrated to the *same* (ε, δ)-DP guarantee—produces models with *drastically different* privacy behaviors.

on *perceptions of model behaviors*, which we term *empirical privacy*,[6] in contrast to the theoretically-grounded definition of DP. The gap between DP's theoretical guarantees and empirical privacy concerns surrounding LLMs has significant implications: research shows that people can understand better the implications of DP than formal definitions [149], and failures to effectively communicate DP's promise can discourage data sharing [150, 151].

In this chapter, we take an initial step toward bridging this gap by investigating the *consistency* of DP with respect to empirical privacy. Specifically, we ask: *Do LLMs calibrated to the same DP guarantee share similar levels of empirical privacy?* To explore this, we fine-tune LLMs using DP-SGD [42, 152, 153] with different hyperparameter configurations, ensuring they achieve the same DP guarantee, and quantitatively assess their empirical privacy through the lens of memorization. Our results reveal a concerning inconsistency, which we refer to as *empirical privacy variance* (Fig. 13).

Our main contributions are as follows: In Sec. 5.2, we formally define our empirical privacy measures and demonstrate the phenomenon of empirical privacy variance, showing it is ubiquitous and substantial, with consistent trends across dimensions such as model, data, and privacy budget (Fig. 14). We further discuss its implications, particularly the challenges it poses for standardization. In Sec. 5.3, we analyze the influence of hyperparameters in

[6]While the term empirical privacy is usually associated with privacy attacks [144, 145, 146] in the literature [147, 148], our definition is broader and more aligned with LLMs: it extends the existing notion by framing vulnerability against attacks as a model behavior and shifts from worst-case threat models to practical, user-focused metrics that reflect tangible privacy risks.

DP-SGD on empirical privacy through regression analyses. Our findings reveal a *no-free-lunch* result: utility gains achieved from hyperparameter tuning often come at the cost of compromised empirical privacy. Based on the insights drawn from hyperparameter analyses, we propose heuristics for hyperparameter selection to improve empirical privacy and demonstrate their effectiveness.

5.1 Preliminaries

We recall the key concepts from Sec. 2.2 and introduce additional notation and background specific to this chapter.

DP-SGD notation. Recall from Sec. 2.2 that DP-SGD operates with a clipping norm c, noise multiplier σ, and mini-batch size b. In this chapter, we additionally use the following *training hyperparameters*: T (number of training iterations) and η (learning rate). We define n as the training set size and $q \coloneqq b/n$ as the sampling rate. We refer to a combination of training hyperparameters as a *configuration*, and an instantiation of DP-SGD with a specific configuration as a *mechanism*. The full algorithms are presented in Appendix C.1.

Memorization in language models. Memorization is a well-documented phenomenon in LLMs [17, 154, 155]. Various notions have been proposed to characterize memorization [134, 156, 157], with recent works further expanding this understanding through concepts like *approximate memorization* [158] and a taxonomy of memorization behaviors [137]. In this work, we use memorization to analyze empirical privacy.

5.2 Landscape of Empirical Privacy Variance

In this section, we demonstrate empirical privacy variance across multiple dimensions and discuss its significance.

Table 1: Example secrets in Enron and TOFU

Dataset	Random samples of secrets
Enron	"Carol St. Clair\nEB 3889\n713-853-3989" "713-853-5620 (phone)\n713-646-3490 (fax)\nsara.shackleton@enron.com"
TOFU	`genre`("Yevgeny Grimkov") ⟶ "cyberpunk" `genre`("Adrianus Suharto") ⟶ "dystopian"

5.2.1 Experimental Setups

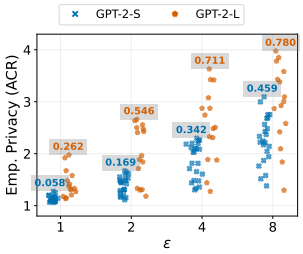
Our experimental framework consists of two main steps: 1) fine-tuning an LLM on a dataset using DP-SGD, and 2) evaluating the empirical privacy (formally defined shortly) of the resulting model. We base our study on two sets of experiments. In the first, we fine-tune GPT-2 models (-small (S) and -large (L); [7]) on Enron Email [159]. In the second, we fine-tune Llama-2 models (-7b and -13b; [160]) on TOFU [161]. We ensure that the fine-tuning examples were not included in the models' pre-training data (see Appendix C.2.5). Below, we introduce the datasets and secrets, DP fine-tuning procedure, and empirical privacy measures.[7]

Datasets and secrets. The Enron Email dataset [159] consists of emails by employees of the Enron Corporation. We perform a series of pre-processing steps including sample-level de-duplication (Appendix C.2.1), resulting in a dataset of 33k samples. We extract small pieces of sensitive information (e.g., phone numbers, see Table 1) from the dataset and define them as the *secrets*. The TOFU dataset [161] contains *synthetic* author profiles describing authors' attributes. We extract the *genre* attribute as the secret (see Table 1 and Appendix C.2.2) as it is relevant and easy to extract and prompt. The secret extraction procedure and the secret statistics are in Appendix C.2.6; we also include a discussion on the privacy unit (Appendix C.2.6).

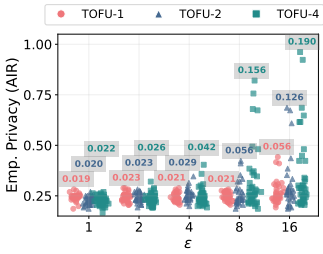
DP fine-tuning. Following prior work [48, 162], we fine-tune LLMs with *LoRA* [163] using DP-SGD/DP-Adam [42, 46], and compute a σ that satisfies a target (ε, δ)-DP guarantee using the PRV accountant [43]. We use common choices of $\varepsilon \in \{1, 2, 4, 8, 16\}$

[7]Our code is publicly available at `https://github.com/empvv/empirical-privacy-variance`.

(a) GPT-2 family models on Enron **Dimension: model** (increased model size)



(b) Llama-2-7b on paraphrase-scaled TOFU **Dimension: dataset** (increased dataset size)



(c) Llama-2-7b on density-adjusted TOFU, $\varepsilon = 8$ **Dimension: dataset** (increased secret density)

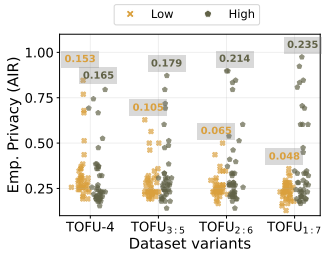


Figure 14: **Empirical privacy variance: ubiquitous, substantial, and revealing intriguing trends.** Each subfigure presents jitter plots of empirical privacy scores (ACR or AIR) obtained by models trained under a given (ε, δ)-DP guarantee. Higher y-axis scores indicate worse empirical privacy, while the x-axis contrasts different groups (e.g., models of varying sizes in (a)), represented by different colors. Within each group, scattered points correspond to unique hyperparameter configurations (b, T, η), averaged over training randomness (we show the impact of training randomness is much smaller than that of hyperparameters in Appendix C.3.3). Each group's *standard deviation* is labeled at the top of its cluster. The subfigures demonstrate that empirical privacy variance increases with **(a)** *model size*, **(b)** *dataset size*, **(c)** *secret density*, and **(a/b)** *privacy budget* ε.

and set $\delta = n^{-1.1}$. Finally, we evaluate the utility of the fine-tuned LLMs on a held-out test set using negative log likelihood (NLL), where lower values indicate better performance. *Hyperparameter choices.* We perform extensive hyperparameter tuning in the space of (b, T, η), while fixing c to a small constant, as we find that varying it within the recommended range [46, 47] has minimal impact on utility or empirical privacy. Following prior work [47, 164], we do not account for the additional privacy loss incurred by hyperparameter tuning on private data [165]. For GPT-2 models on Enron, we perform a *partial* hyperparameter sweep, resulting in 23 configurations for GPT-2-S and 15 for GPT-2-L. For Llama-2 models on TOFU, we conduct a *full* grid search over b, T, η, yielding 60 configurations per setting. The difference between partial and full sweeps is due to compute constraints (see Appendix C.2.9). Each configuration is fine-tuned with multiple random seeds, and we retain models achieving at least 90% of the utility *gain* from the pre-trained baseline to the best-performing model. Further details on fine-tuning are deferred to Appendix C.2.9.

Empirical privacy measures. In this chapter, we focus on a pragmatic view of

privacy based on the perceptions of model behaviors, i.e., memorization and regurgitation of secrets. Specifically, we quantify empirical privacy through the following *memorization* scores. Let M be a mapping from the input/prompt to the output/generation produced by greedy or stochastic decoding on the model.

On Enron, let s denote a secret string. We consider:

- Adversarial compression ratio (ACR; [157]) measures how effectively a secret is stored in model weights, by optimizing for the shortest prompt eliciting it:

$$\text{ACR}(s) = \frac{|s|}{|p^*|}, \text{ where } p^* := \arg\min_p |p| \text{ s.t. } M(p) = s.$$

- Verbatim memorization ratio [VMR; adapted from 134] evaluates whether prompting with the prefix (s_1) of a secret leads to recovery of the remainder (s_2):

$$\text{VMR}(s; s_1, s_2) = \mathbb{1}[M(s_1) = s_2], \text{ where } s = s_1 \| s_2.$$

On TOFU, let x be an author, $A(x)$ the author's attribute (genre), and $\mathcal{P}(x)$ a prompt aiming to elicit the secret ("What genre does $\{x\}$ write in?"). We consider:

- Attribute inference ratio (AIR; our proposed metric) measures the model's ability to recover a secret attribute in response to a prompt query:

$$\text{AIR}(x) = \mathbb{1}[A(x) \text{ appears in } M(\mathcal{P}(x))].$$

We compute the average of each of these metrics (ACR, or VMR, or AIR) over a curated set of secrets, and refer to them as *empirical privacy scores*. Higher scores correspond to stronger memorization and weaker empirical privacy. Empirical privacy variance is defined as the variance of these scores in each controlled setting. Additional details about these metrics are provided in Appendix C.2.7.

5.2.2 Trends and Generality of Empirical Privacy Variance

Fig. 14 reveals *substantial* empirical privacy variance among high-utility models for commonly adopted ε values. For instance, a Llama-2-7b trained on TOFU-4 at $\varepsilon = 8$ can either nearly fully reveal the secrets (AIR higher than 0.8) or have little knowledge of them (Fig. 14(b)). We proceed to investigate empirical privacy variance across different dimensions.

Trends. We analyze the trends of the variance across key dimensions in DP fine-tuning of LLMs. *Model*: Fig. 14(a) shows that empirical privacy variance increases with model size (from 117M to 774M). *Data*: Fig. 14(b-c) focuses on the influence of data. We generate TOFU variants with different dataset size and secret density. *Paraphrase-scaled TOFU* (TOFU-2, TOFU-4) expands the original dataset by 2× and 4× via paraphrasing (Appendix C.2.4). *Density-adjusted TOFU* applies non-uniform augmentation to two randomly partitioned groups, yielding 1:7, 2:6, 3:5 size ratios. TOFU-4 (4:4) serves as a uniform-density reference of the same size. Fig. 14(b-c) show that larger dataset or higher secret density leads to larger variance.

Privacy budget: Fig. 14(a-b) demonstrate a consistent trend: empirical privacy variance increases with ε.

Fine-tuning paradigm: Full fine-tuning yields higher variance than LoRA, as we show in Appendix C.3.1.

Generality. To demonstrate the generality of these trends, we examine two additional dimensions: *secret subsets* and *empirical privacy measures*. Fig. 15 shows that across these dimensions, empirical privacy variance increases with ε, dataset and model size, aligning with the trends observed in Fig. 14. More results are deferred to Appendix C.3.1.

Intuition. The positively contributing factors (larger models, larger paraphrased datasets, higher secret density, larger ε) all intuitively lead to stronger memorization [134, 158]. This intuition is empirically confirmed by our results as well, which show increasing *average* empirical privacy scores. However, a more fundamental trend we uncover is the

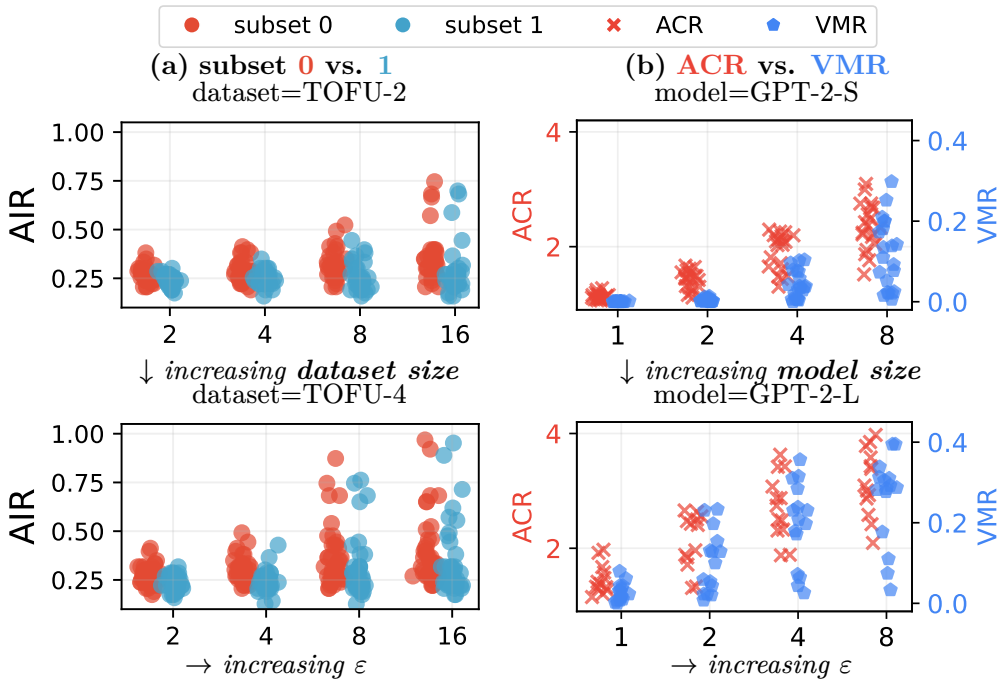


Figure 15: **Generality of empirical privacy variance.** Across **(a)** *secret subsets* (subset **0 vs. 1**) and **(b)** *empirical privacy measures* (**ACR vs. VMR**), we observe consistent trends as in Fig. 14: empirical privacy variance increases with ε ($\rightarrow$ in each subfigure), dataset size ($\downarrow$ in column (a)), and model size ($\downarrow$ in column (b)).

rise in empirical privacy *variance*. We note this is a novel phenomenon and less intuitive than the increase in average scores. We defer further discussions to Appendix C.3.2.

5.2.3 Discussion

Why is this surprising? It is well-known that the interpretation of a DP guarantee heavily depends on the context: even under the same (ε, δ)-DP guarantee, variations in factors like data characteristic [e.g., real-world vs. adversarially constructed, 166], model architecture [e.g., ResNet vs. CNN, 167], and training algorithm [e.g., full vs. LoRA fine-tuning, 168, 169] can lead to different privacy implications. In contrast, we control for these factors and further restrict to models with good utility (thus avoiding trivial cases like zero updates). Despite this control, we observe substantial empirical privacy variance, highlighting the under-explored role of hyperparameters.

Why is this relevant? Consider classic DP mechanisms such as the Laplace and

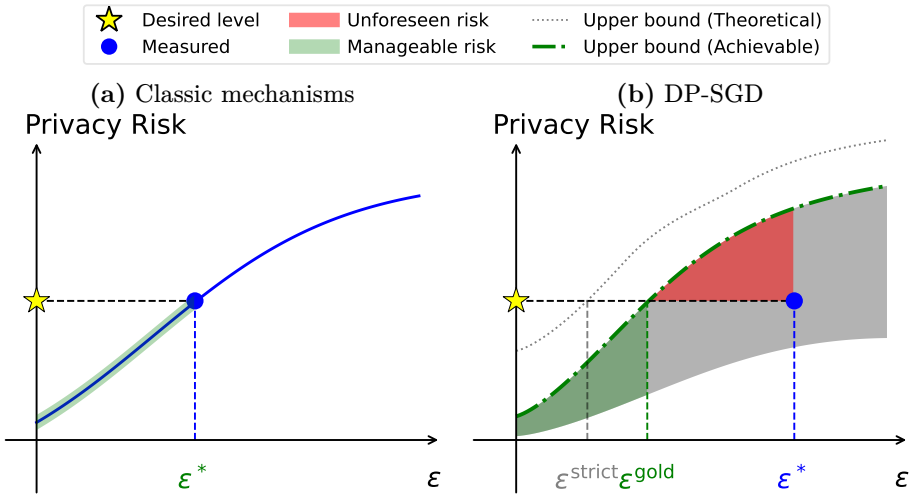


Figure 16: A *conceptual illustration* of classic mechanism vs. DP-SGD. In classic mechanisms, the monotonic relationship between privacy risk and privacy budget ε allows any $\varepsilon \leq \varepsilon^*$ to be certified if ε^* satisfies the desired privacy risk. In DP-SGD, however, variance introduces an *achievable region* of privacy risk, reflected by the upper and lower bound. A measured configuration meeting the privacy requirements does not safeguard the corresponding ε^*; identifying the truly reliable threshold, $\varepsilon^{\text{gold}}$, requires testing a wide range of configurations to account for the full spectrum of privacy risks. While a conservative *theoretical* upper bound [170, 171, 172] could aid in standardization by identifying $\varepsilon^{\text{strict}}$, such bounds are generally unavailable for empirical privacy measures like ACR.

Gaussian mechanisms [173]. Their noise parameter (scale parameter b for Laplace and σ for Gaussian) inversely correlates with ε and uniquely determines privacy risk: increasing it lowers the signal-to-noise ratio, making it harder for adversaries to extract meaningful information. This establishes a one-to-one, monotonic ε-to-risk relationship. In contrast, the *composition* nature of DP-SGD results in a one-to-many ε-to-risk relationship, making ε insufficient to fully capture privacy risk.

A direct consequence is that, in DP-SGD, ε cannot be used for *certification*: a model calibrated to a given ε^*, deemed to meet privacy requirements, cannot ensure compliance for models with stricter DP guarantees ($\varepsilon \leq \varepsilon^*$). This limitation further complicates *standardization*, i.e., establishing an ε^* for practitioners to follow. If a legislative body runs privacy tests (independent of ϵ) and recommends ε^* as a privacy standard without accounting for empirical privacy variance, there will be unforeseen risks that undermine the efficacy of such a standard (see Fig. 16 for an illustration).

5.3 How Hyperparameters Impact Empirical Privacy: Analysis and Selection Heuristics

In this section, we analyze the impact of hyperparameters through regression analyses, based on which we reveal a no-free-lunch result for empirical privacy and propose refined heuristics for hyperparameter selection. Although a linear model might not fully capture the complex relationship between empirical privacy scores and hyperparameters, we mainly use it as an exploratory tool to gain *qualitative* insights rather than to deliver definitive quantitative conclusions.

5.3.1 Dissecting Effects of Hyperparameters

We use `lm()` in R Statistical Software (v4.4.2) [174] to perform multivariate regression, where the target y is the empirical privacy score and the covariates are the hyperparameters b, T, η. Regression is conducted in logarithmic space (log-transforming the covariates) to

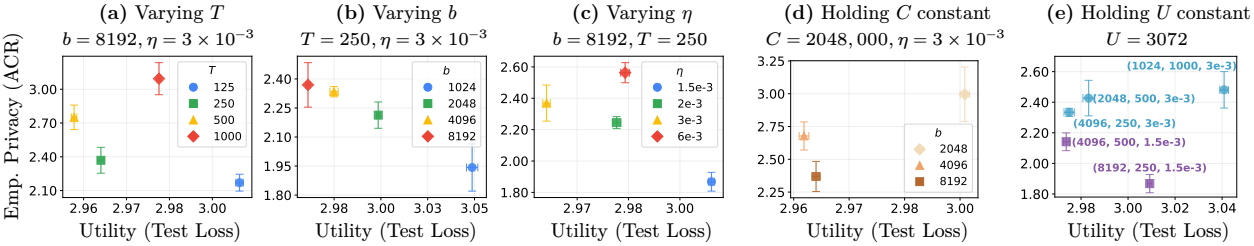


Figure 17: **Effect of individual and composite hyperparameters** (setting: GPT-2-S, Enron, ACR, $\varepsilon = 8$). We show the empirical privacy and utility of the DP fine-tuned models using different hyperparameters. **(a-c)**: Varying one hyperparameter while holding the others fixed. **(d)**: Holding compute ($C = b \cdot T$) fixed and varying (b, T); **(e)**: Holding updates ($U = C \cdot \eta$) fixed and varying (C, η).

Table 2: (a) Regression on *individual* hyperparameters

	Enron		TOFU	
Variable	Coef.	p-value	Coef.	p-value
Batch size ($\log b$)	0.13^{***}	1×10^{-5}	0.029^{***}	2×10^{-5}
Iterations ($\log T$)	0.37^{***}	$< 2 \times 10^{-16}$	0.048^{***}	1×10^{-11}
Learning rate ($\log \eta$)	0.51^{***}	5×10^{-15}	0.068^{***}	3×10^{-12}

(b) Regression on *composite* hyperparameters

	Enron		TOFU	
Variable	Coef.	p-value	Coef.	p-value
Compute ($\log C$)	0.22^{***}	2×10^{-12}	0.039^{***}	5×10^{-11}
Learning rate ($\log \eta$)	0.53^{***}	6×10^{-13}	0.066^{***}	3×10^{-11}

Notes: $^{***}p < 0.001$, $^{**}p < 0.01$, $^{*}p < 0.05$. The response variable (empirical privacy score y) is ACR for Enron and AIR for TOFU, leading to different scales of the coefficients, as ACR and AIR have different ranges.

examine the impact of *multiplicative* changes to each hyperparameter. We focus on two settings: DP fine-tuning GPT-2-S on Enron at $\varepsilon = 4$ and Llama-2-7b on TOFU at $\varepsilon = 16$. The total number of instances for regression is 92 and 114, respectively.

Regression on individual hyperparameters. We regress y on $(\log b, \log T, \log \eta)$. The results are presented in Table 2. The p-values and the coefficients indicate a statistically significant positive relationship between individual hyperparameters and empirical privacy. Additionally, the coefficient for $\log b$ is the smallest, while $\log \eta$ has the largest.

Regression on composite hyperparameters. We analyze the interactions between individual hyperparameters and their joint effects. Specifically, we combine b and T into

a composite quantity called **compute** $C \coloneqq b \cdot T$ (while retaining η as a separate term due to its large coefficient), which represents the total training effort—a key concept in neural scaling laws [9, 10]. Additionally, we define **updates** $U \coloneqq C \cdot \eta$, representing the total cumulative learning signal during training. These composite hyperparameters, along with the individual hyperparameters, form a hierarchy (see Fig. 18). We regress y on $(\log C, \log \eta)$. Table 2 shows that the coefficient of $\log C$ is much smaller than that of $\log \eta$.

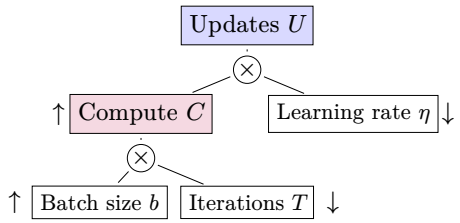


Figure 18: Hierarchy of hyperparameters. Arrows indicate the direction to improve empirical privacy when the parent node is fixed.

Interpretations of regression results.

- *Positive coefficients*: Increasing any individual hyperparameter worsens empirical privacy.
- *Batch size in compute*: For fixed compute ($C = b \cdot T$), increasing b (while decreasing T proportionally) improves empirical privacy due to the smaller coefficient of $\log b$ (e.g., doubling b while halving T has a smaller net effect).
- *Learning rate in updates*: For fixed updates ($U = C \cdot \eta$), decreasing η (while increasing C proportionally) improves empirical privacy.

Case studies. We validate the above interpretations through case studies. For individual hyperparameters, we fix two and vary the third, observing that empirical privacy deteriorates as T, b, or η increases (Fig. 17(a-c)). For batch size in compute, we analyze configurations with the same compute and learning rate, showing that a larger b improves empirical privacy (Fig. 17(d)). Similarly, for learning rate in updates, among configurations with the same updates, we find that a smaller η yields better empirical privacy (Fig. 17(e)). These findings support our interpretations.

5.3.2 Improving Hyperparameter Selection

Existing practices. Our findings in Sec. 5.3.1 reveal a no-free-lunch result in empirical privacy. Previous practices of hyperparameter tuning in DP-SGD focus on optimizing utility under a fixed ε, recommending larger batch size [46, 138, 164], higher learning rate (at larger batch size) [175], and more training iterations [164, 176]. While these recommendations do lead to better utility (see Fig. 17), Sec. 5.3.1 shows they also compromise empirical privacy. This highlights that *the gains in utility come at the expense of empirical privacy*, challenging the conventional notion of "utility-privacy trade-off" that largely focuses on the utility-ε trade-off but neglects empirical privacy. We argue that evaluating DP mechanisms requires incorporating empirical privacy as a third dimension, alongside utility and ε, for a more comprehensive assessment.

Refined heuristics. Given the limitations of existing hyperparameter tuning practices, we propose refined heuristics for hyperparameter selection that explicitly account for empirical privacy. Building on the insights from Sec. 5.3.1, we describe a set of *pairwise comparison* heuristics:

Takeaway: A configuration (b_1, T_1, η_1) is expected to demonstrate better empirical privacy than an alternative (b_2, T_2, η_2), if either:

1. **Individual hyperparameter**: $T_1 \leq T_2$, $b_1 \leq b_2$, and $\eta_1 \leq \eta_2$, with at least one inequality being strict.
2. **Compute**: $C_1 = C_2$, $\eta_1 = \eta_2$, and $b_1 > b_2$.
3. **Updates**: $U_1 = U_2$, and $\eta_1 < \eta_2$.

We comment that, defining and measuring empirical privacy is challenging as it depends on the task and use case. In this regard, our heuristics are useful in that they allow practitioners to compare and select configurations likely to demonstrate good empirical privacy *without measuring it.* Nevertheless, in domains with known and well-defined privacy risks [177, 178], we strongly encourage placing application-specific upper bounds [179]

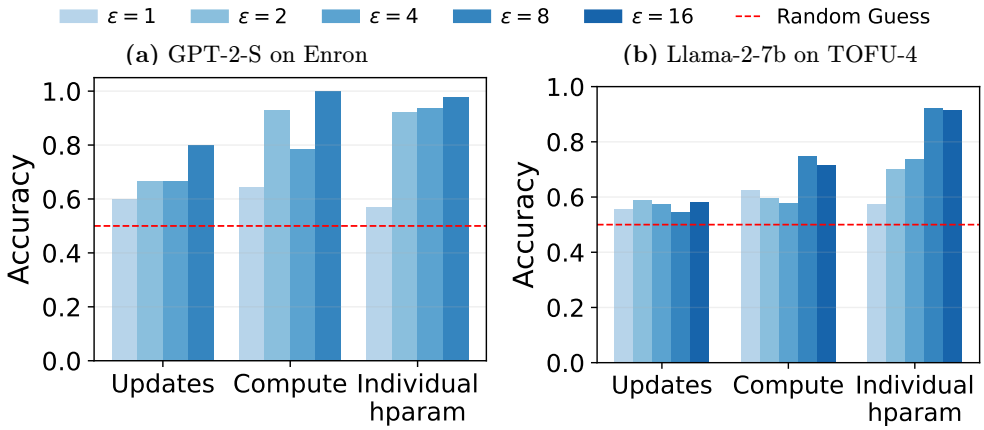


Figure 19: Accuracy of three heuristics in two settings across ε's.

on both formal DP guarantees and empirical privacy measurements, and then tuning hyperparameters to stay within those bounds.

Accuracy of heuristics. We define the accuracy of a heuristic as the proportion of correct predictions among pairs that satisfy the condition. For example, consider the *compute* heuristic. In the log-space hyperparameter cube (Fig. 20), relevant pairs lie on the anti-diagonals with the same color (so $C = b \cdot T$ is constant) of each $\log b$-$\log T$ plane.

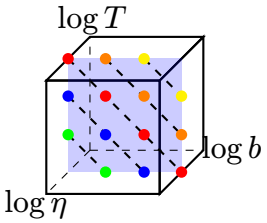


Figure 20: Hyperparameter cube in log space.

Fig. 19 shows the accuracy achieved by each heuristic across two settings. The proposed heuristics significantly outperform the random guess baseline. Importantly, the heuristics **generalize** beyond the two scenarios that they are developed from (Sec. 5.3.1), e.g., to different ε's. We refer the readers to Appendix C.4.1 for a comprehensive set of results across all settings.

5.3.3 Practical Evaluation of Proposed Heuristics

Beyond evaluating pairwise comparison accuracy, we assess the *usefulness* of the heuristics in a real-world application: *selection among a pool of candidate models.*

Objective. Given a pool of models satisfying an (ε, δ)-DP guarantee and a minimum utility threshold u, the goal is to select a model (referred to as a "point" hereafter) with strong empirical privacy. We denote the point with the optimal empirical privacy score as the *oracle* point. See Fig. 21 for an illustration.

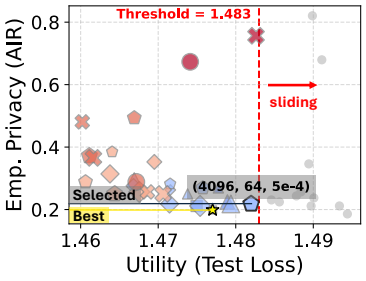


Figure 21: Each point corresponds to a model/configuration. Red dashed line: utility threshold u, defining a *subpool* of points P_u to its left. Yellow star: *oracle* point. Setting: Llama-2-7b, TOFU-4, $\varepsilon = 8$.

Our procedure and baselines. We propose a *sequential* hyperparameter selection procedure in Alg. 2 based on the three heuristics derived in Sec. 5.3.2. Following the hierarchy in Fig. 18, Alg. 2 applies these **heuristics** from top to bottom, discarding points at each step. If multiple points remain, we further leverage a *worst-utility* **heuristic** to break ties, based on the common belief of utility-privacy trade-off. We compare our procedure to two baselines: the usual practice of selecting the *best-utility* point, and a *standalone worst-utility heuristic.*

Evaluation. We slide a utility threshold u from the leftmost to the rightmost (Fig. 21). At each point it crosses with utility u, we evaluate the selection methods on the subpool of points P_u to the left of the threshold and compute their *relative privacy risks*, defined as the relative difference in empirical privacy scores between the selected and oracle points: $\left(y_{\text{selected}(P_u)} - y_{\text{oracle}(P_u)}\right) / y_{\text{oracle}(P_u)}$. We report the average of the relative privacy risk over all u's.

Algorithm 2: Procedure of hyperparameter selection

Require: A set of points $\mathcal{P} = \{(b, T, \eta)\}$
Ensure: A single selected point (b^*, T^*, η^*)

1: **Step 1 (Updates heuristic):** Group $\mathcal{P}$ by U, and retain points with the minimal η in each group.
2: **Step 2 (Compute heuristic):** Group the remaining points by (U, C), and retain points with the maximal b in each group.
3: **Step 3 (Individual hyperparameter heuristic):** Among the remaining points, discard any point (b_1, T_1, η_1) if there exists another point (b_2, T_2, η_2) such that $b_1 \geq b_2$, $T_1 \geq T_2$, $\eta_1 \geq \eta_2$, and at least one inequality is strict.
4: **Final step (Worst-utility heuristic):** From the remaining points, select the one with the worst utility and return it.

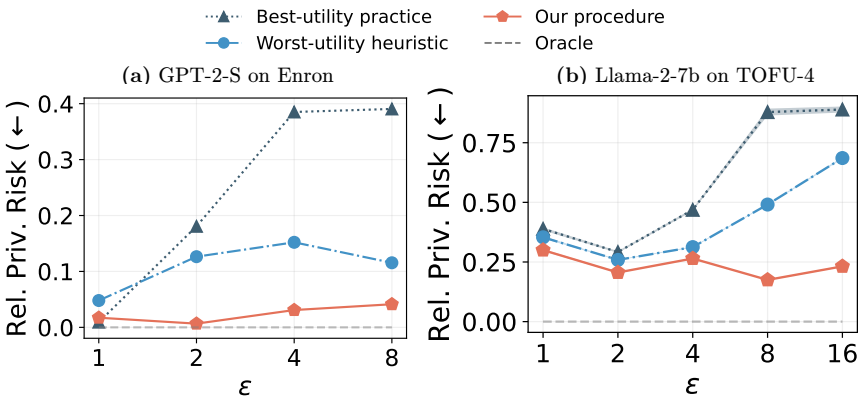


Figure 22: Relative privacy risk of our procedure compared with baselines in two settings across varying ε.

Results. In Fig. 22, we compare the relative privacy risks of the selection methods across two settings with varying ε's. Our proposed procedure *consistently* outperforms the baselines by a large margin. These results not only validate the effectiveness of the underlying heuristics but also highlight their ability to **generalize**. Additional results in Appendix C.4.3 further confirm this generalization across different models and datasets.

5.4 Related Work

Two recent works study how different mechanisms with the same DP guarantee can yield varying privacy implications. Hayes et al. [172] show that increasing q or T boosts the success rate of reconstruction attacks. Kaissis et al. [180] propose *approximate Blackwell dominance* to compare mechanisms sharing the same (ε, δ), which quantifies the *maximum excess vulnerability* when choosing one mechanism over another. They find vulnerability increases with q or T. Our findings align with theirs: increasing q or T degrades empirical privacy. This indicates that the phenomenon is general, likely driven by fundamental factors, and points to an intriguing avenue for future research.

While they primarily rely on theoretical analyses and worst-case threat models, our study focuses on the practical setting of *language model fine-tuning*, highlighting real-world risks. We provide a fine-grained analysis of how individual (including η, absent in prior work) and composite hyperparameters influence empirical privacy and offer *heuristics* for hyperparameter selection that accounts for empirical privacy. Taken together, our work reinforces prior findings, extends them in new directions, and offers actionable insights for researchers, practitioners, and policy-makers.

5.5 Conclusion

This work reveals empirical privacy variance—models calibrated to the same (ε, δ)-DP guarantee using DP-SGD with different hyperparameters exhibit significant variations in their empirical privacy. We believe this work marks a crucial initial step towards bridging

the gap between theoretical and empirical privacy in LLMs and beyond.

Chapter 6 Differentially Private Conditional Text Generation with RL-Boosted Control

This chapter moves beyond the study of the gap between formal and empirical privacy to a downstream objective of central importance: generating high-quality synthetic text under differential privacy. Progress in generative AI depends not only on stronger models, but also on expanding the range of data resources that can be used safely and effectively. Differentially private synthetic data [33] is a particularly promising direction, as it offers a reusable and privacy-preserving way to unlock the value of sensitive user data that would otherwise remain inaccessible. Using the DP fine-tuning tools introduced in 2.2, this chapter proposes a hierarchical framework for differentially private synthetic text generation that separates structured feature learning from conditional text generation, and augments it with reinforcement learning to improve the controllability of the generator.

Modern AI applications rely on vast amounts of user data, ranging from keyboard inputs on mobile devices [49, 181, 182] and recommender interaction histories [183, 184] to conversational preferences [11, 12]. This reliance poses significant privacy risks, which have become especially pressing with the rise of large language models (LLMs), as recent studies show they can memorize sensitive information from training corpora and expose it during user interactions [17, 135, 155].

To maximize the value of data while preserving user privacy, a promising approach is to generate differentially private (DP) synthetic data [33]. This paradigm offers a key advantage over task-specific DP mechanisms: it avoids the cumbersome need to design a new solution for each application. Instead, the DP synthetic dataset can be reused across any downstream task without incurring additional privacy cost or requiring changes to existing data pipelines. DP synthetic text, in particular, has attracted growing interest, spurring a line of work that harnesses the power of LLMs through fine-tuning or API-based prompting to continually push the frontier of the privacy-utility trade-off [185, 186, 187, 188, 189, 190, 191].

Despite these advances, most existing work on DP synthetic text remains limited to producing synthetic *datasets*, overlooking the critical need for *conditional generation* [192, 193]. Conditional generation offers flexibility by enabling fine-grained control over the generation process, a capability of significant practical value. It allows users to synthesize data tailored to specific requirements (e.g., emails with positive sentiment) and enables analysts to preserve key statistical attributes or ensure balanced representation across subpopulations through controlled variations [194]. Moreover, as we will see shortly, conditional generation enables us to obtain not only private synthetic text but also private synthetic *features*, which can be valuable for a wide range of downstream analytical tasks.

In this work, we develop an integrated approach to DP (conditional) text generation that achieves high-quality synthetic text and fine-grained control under strong privacy guarantees. Specifically, our contributions are as follows:

- **A hierarchical framework for DP synthetic text generation.** We propose a framework that decomposes the problem of generating DP synthetic text into two subtasks: feature learning and conditional text generation. This modularity allows for systematic optimization, and through comprehensive ablations, we identify the most effective configuration: a rich tabular schema for feature, a specialized DP tabular synthesizer, and a DP fine-tuned conditional generator, which we collectively term **ACTG**[8] (**A**ttribute-**C**onditioned **T**ext **G**eneration).
- **Boosting fine-grained control in ACTG.** While ACTG produces high-quality synthetic datasets, we observe that its conditional generator suffers a significant loss in instruction-following ability under DP. To address this, we develop **Anchored RL** (ARL), a post-training recipe applied on top of ACTG. It combines a reinforcement learning (RL) objective to improve control with a supervised fine-tuning (SFT) objective on best-of-N data, anchoring the model to the private text distribution and mitigating reward hacking.

[8]The acronym reflects a biological metaphor: as DNA is built from four bases (A, C, T, G), *feature* is the basic unit of our conditional generation framework.

- **State-of-the-art results in DP conditional text generation.** On challenging, real-world datasets, our integrated approach, **ACTG-ARL**, which combines ACTG with ARL, establishes a new state of the art. It *simultaneously* advances the quality of DP synthetic text over prior work (+20% in MAUVE and +50% in attribute distribution matching) while delivering a conditional generator with strong instruction-following capabilities. This enables fine-grained, controllable generation for diverse and practical applications.

6.1 Preliminaries

We review the basics of DP synthetic data for both text and tabular domains. For the definition of differential privacy and DP-SGD, we refer the reader to Sec. 2.2.

DP synthetic data. DP synthetic data [195, 196] provides a privacy-preserving surrogate of the original dataset, aiming to retain core utility under formal DP guarantees. Due to the post-processing property of DP [173], such data can be freely shared and used for downstream tasks without additional privacy cost.

Text data. Since the seminal work of Yue et al. [185], DP fine-tuning (DP-FT) has become the dominant approach for generating DP synthetic text [186, 189, 191]. In this approach, a base model is fine-tuned on private text using a DP optimizer such as DP-Adam [46]. While Private Evolution (PE) [187, 197] has emerged as a promising alternative, recent evidence indicates that DP-FT on a moderately sized model already outperforms PE [191]. We defer more discussion of related work to Sec. 6.4.

Tabular data. Generating high-dimensional synthetic tabular datasets under DP is a well-studied problem, with strong algorithms (e.g., AIM [198]) and standardized benchmarks [199, 200]. A key design choice of our framework (Sec. 6.2.2) is to build on this foundation by recasting part of the synthetic text problem as a tabular synthesis task.

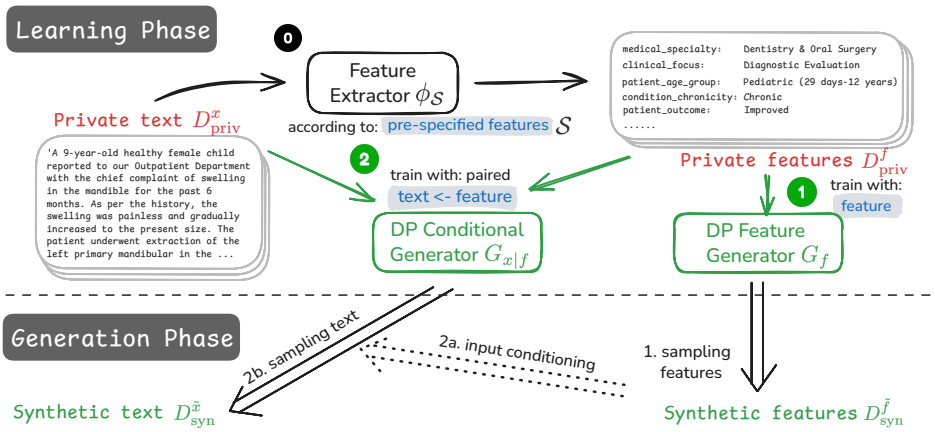


Figure 23: Our hierarchical framework for DP synthetic text generation.

6.2 A Hierarchical Framework for DP Synthetic Text Generation

In this section, we introduce a modular, hierarchical framework for DP synthetic text generation. We detail algorithmic design choices and present a comprehensive empirical evaluation to verify its effectiveness and ablate core components.

6.2.1 A Hierarchical Framework

Our framework generalizes CTCL [191], which fine-tunes a conditional generator using *topics* as input, guided by a privatized topic histogram derived from a pretrained topic model. Compared to learning the private text distribution end-to-end, CTCL enjoys two advantages: i) histograms have low sensitivity and thus retain high utility under a tight privacy budget, and ii) conditioning simplifies the synthesis task, as topics guide the generator. These factors together yield a favorable privacy-utility trade-off and state-of-the-art results in *resource-constrained* DP synthetic text generation.

However, CTCL has several limitations that undermine its reliability in some settings. It relies on a fixed topic model trained on a public corpus, which may not align with the private domain and can force nuanced text into coarse, lossy categories, making the inferred topics inaccurate. Moreover, when the dataset size is small relative to the number of topics, the topic histogram contains many empty bins, yielding a low signal-to-noise

ratio that renders the privatization step unstable (see Appendix D.4.10 for analysis). The combination of inaccurate topic inference and noise-dominated histograms can limit the utility of the generated text. This motivates us to generalize CTCL into a broader and more flexible framework and systematically analyze its design choices. We propose a modular, hierarchical framework that *decomposes* DP synthetic text generation into two subtasks: learning a *low-dimensional* feature representation of private text, and learning a conditional generator. Given a feature design $\mathcal{S}$, Fig. 23 outlines the **Learning Phase**:

- *Stage 0: Feature extraction.* We employ a feature extractor $\phi_S : x \to f$ to extract the private feature set D^f_{priv} from the raw text corpus D^x_{priv} according to S, where x denotes a text sample and f its feature. We also compose the (feature, text) pairs into $D^{f,x}_{\text{priv}}$. This pre-processing step prepares the data for subsequent stages.
- *Stage 1: Learning a DP feature generator.* We learn a generator G_f with privacy budget ε_1 on D^f_{priv}. The goal is to generate synthetic features that resemble the private ones.
- *Stage 2: Learning a DP conditional generator.* We learn a conditional generator $G_{x|f}$ with privacy budget ε_2 on $D^{f,x}_{\text{priv}}$. The goal is to generate synthetic text that resembles the private text, while adhering to the requirements specified by the features.

In the **Generation Phase**, once G_f and $G_{x|f}$ are obtained, DP synthetic text can be produced without further access to private data. We first sample synthetic features from the DP feature generator $\tilde{f} \sim G_f$, and then feed them into the DP conditional generator to produce the final output $\tilde{x} \sim G_{x|f}(\cdot \mid \tilde{f})$. We refer to the synthetic DP feature set as $D^{\tilde{f}}_{\text{syn}}$ and the synthetic DP text set as $D^{\tilde{x}}_{\text{syn}}$.

Remark 6.1. Stage 0 (feature extraction) is a pre-processing step and does not consume any privacy budget (we treat the feature extractor as a trusted component and defer more discussions to Appendix D.1.1). The overall privacy guarantee of our framework is (ε, δ), obtained by composing the budgets of Stage 1 (ε_1) and Stage 2 (ε_2). We rely on advanced composition for privacy accounting and defer detailed descriptions to Sec. 6.2.3 and Appendix D.1.2.

Remark 6.2. CTCL [191] is an instantiation of our framework: the feature $\mathcal{S}$ is *topic*, the feature extractor $\phi_{\mathcal{S}}$ is a *topic model*, the DP feature generator is a *privatized topic histogram*, and the DP conditional generator is a *DP fine-tuned language model* pretrained on a public dataset.

Remark 6.3. A natural alternative to our hierarchical framework is a single-stage conditioning approach that learns to generate the concatenation of (f, x). We evaluate this baseline and find our framework consistently stronger; see Appendix D.4.2.

6.2.2 Instantiation of the Framework

While CTCL represents one concrete instantiation, our framework's modularity defines a rich design space with a wide range of algorithmic choices. We exploit this flexibility to conduct comprehensive ablation studies and identify an optimal configuration. In what follows, we explore this design space across three key dimensions:

Feature design and extraction. We explore three distinct feature designs ($\mathcal{S}$), each requiring a specialized feature extractor ($\phi_{\mathcal{S}}$):

($\mathcal{S}_1$) **Topic** (CTCL, Tan et al. [191]): A single topic defined by a list of keywords, extracted using the pretrained topic model from their work ($\phi_{\mathcal{S}_1}$)[9].

($\mathcal{S}_2$) **Free-form summary:** A concise summary of the text with 1-2 sentences, generated by a powerful LLM M_{oracle}, which serves as the extractor ($\phi_{\mathcal{S}_2}$).

($\mathcal{S}_3$) **Structured tabular schema:** A rich, multi-attribute schema with fixed options per attribute, treated as tabular data and annotated by M_{oracle} ($\phi_{\mathcal{S}_3}$). Unlike a set of fixed, generic topics, the schema is *dataset-specific* and designed to capture the key dimensions of the data. We describe the LLM-assisted process for schema design and feature extraction in Appendix D.1.1.

DP feature generator (G_f). The choice of feature generator depends on the feature design. For $\mathcal{S}_1$, CTCL used a privatized histogram. For $\mathcal{S}_2$ and $\mathcal{S}_3$, we consider two types

[9]Their pretrained topic model can be downloaded from `https://github.com/tanyuqian/synthetic-private-data`

of feature generator: DP-FT [185] and AIM [198]. DP-FT applies to any feature that can be represented in a textual format, including free-form and schema-based features, whereas AIM is tailored to tabular data with categorical or numerical features.

DP conditional generator ($G_{x|f}$). For conditional text generation, we consider two approaches: 1) performing DP-FT on a base LLM on the paired (feature, text) set $D^{f,x}_{\text{priv}}$, such that it learns to generate synthetic text conditioned on the input feature, or 2) prompting a powerful LLM (M_{gen}), leveraging its strong instruction-following capabilities.

6.2.3 Experiments

We conduct a comprehensive empirical evaluation to demonstrate the effectiveness of our framework against strong baselines and assess the impact of its core components through detailed ablations. We first describe our experimental setup (Sec. 6.2.3) and then present and discuss the results (Sec. 6.2.3).

Experimental setup Datasets. We conduct experiments on two challenging, domain-specific datasets: **bioRxiv** [190], a corpus of scientific abstracts ($n = 29$k, average number of tokens per sample around 300), and **PMC-patients** [201], a collection of sensitive clinical notes ($n = 240$k, average tokens per sample around 450). These represent a more difficult testbed than the general-domain corpora (e.g., Yelp [202]) frequently used in prior work [185, 187, 191]. We provide further details for both datasets in Appendix D.3.1.

Baselines. We compare against three baselines: Aug-PE [187], vanilla DP-FT [185], and CTCL [191]. While DP-FT has been reported as a weak baseline when implemented with GPT-2 [7], we find its performance improves substantially with stronger base models, consistent with observations in Kurakin et al. [186] and Yu et al. [189]. CTCL (corresponding to $\mathcal{S}_1$) was originally studied under resource-constrained settings, and we adapt it to our setup with a larger model, though without additional pretraining. Finally, we refer to the two proposed designs, $\mathcal{S}_2$ and $\mathcal{S}_3$, as *our conditional generation approaches.* **Implementation**

details. For fair comparison, we use the same base model `gemma-3-1b-pt`[10] [203] for all methods that require fine-tuning (vanilla DP-FT, CTCL, and our approaches); for Aug-PE, we instead use powerful instruction-tuned models `Qwen2.5-7B-Instruct`[11] [204] and `gemini-2.5-flash-lite`[12] (Appendix D.4.9). We use `gemini-2.5-flash-lite`[12] also as the oracle model for feature extraction (M_{oracle}) and as the conditional generator for prompting (M_{gen}). For completeness, we also experiment with a larger base model `gemma-3-4b-pt` (Appendix D.4.4), as well as an open-source model `Qwen2.5-32B-Instruct` as M_{oracle} (Appendix D.4.5). Further implementation details are in Appendix D.3.2.

Privacy budget and accounting. We evaluate all methods under three total privacy budgets: $\varepsilon \in \{1, 4, \infty\}$. Following standard practice [185, 187], we set $\delta = 1/(n \log n)$, where n is the size of the private training set. The total privacy cost of our framework is the composition of the budgets for the feature generator (ε_1) and the conditional generator (ε_2). For each method and each total budget ε, we independently tune the budget split $(\varepsilon_1, \varepsilon_2)$. Full details on our privacy accounting are in Appendix D.1.2.

Evaluation suite. Our evaluation suite assesses data quality along multiple dimensions, capturing both broad and specific aspects of synthetic text. First, for general *fidelity*, we follow prior work [185, 187] and use MAUVE [205] to quantify semantic similarity between the synthetic and private text distributions. For *utility*, we measure the F1 score on a downstream classification task, and the next token prediction (NTP) accuracy on a downstream generation task. Finally, we evaluate fine-grained *attribute distribution matching* according to the features in $\mathcal{S}_3$. Specifically, we define d^f_{JS} as the Jensen-Shannon distance between the private and synthetic feature distributions (extracted from D^x_{priv} and $D^{\tilde{x}}_{\text{syn}}$), averaged over all attributes considered. This metric captures discrepancies in feature distributions that matter for downstream analysis, reflecting what an analyst might care about in practice. In Appendix D.4.11, we further evaluate topic distribution matching,

[10] `https://huggingface.co/google/gemma-3-1b-pt`
[11] `https://huggingface.co/Qwen/Qwen2.5-7B-Instruct`
[12] `https://ai.google.dev/gemini-api/docs/models#gemini-2.5-flash-lite`

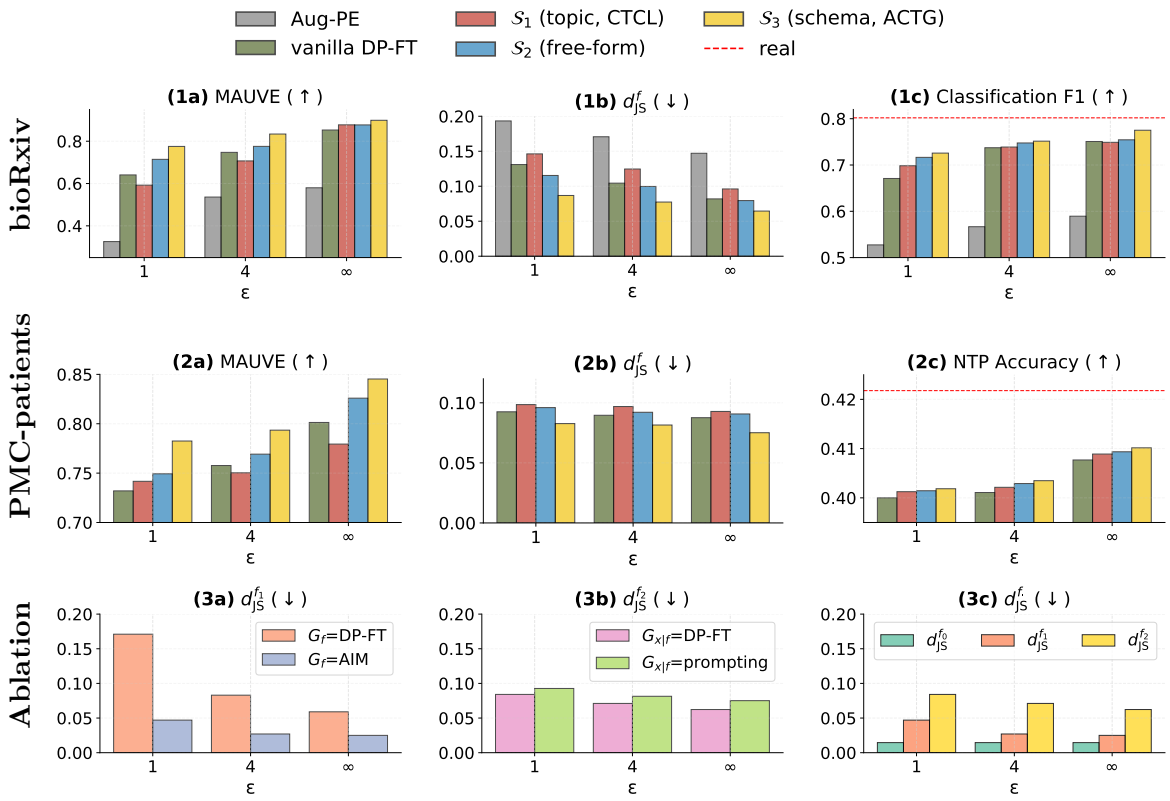


Figure 24: **End-to-end and modular evaluation of our hierarchical framework. (Rows 1–2)** End-to-end comparison of our approaches with baselines (Aug-PE, vanilla DP-FT, CTCL) on bioRxiv and PMC-Patients, evaluated on fidelity (MAUVE, d_{JS}^{f}) and utility (classification F1, NTP accuracy). We omit Aug-PE on PMC-Patients in Row 2 (see full results in Appendix D.4.8) as its performance is substantially lower than other methods. For utility evaluation, we report mean and standard deviation over three trials in Appendix D.4.6. **(Row 3)** Modular ablations and fine-grained error analysis for $\mathcal{S}_3$. Arrows in the figure titles indicate whether higher (↑) or lower (↓) values are better.

which goes beyond schema attributes to assess the alignment in broad topic structure. Detailed descriptions and implementations of all metrics are provided in Appendix D.3.3.

Remark 6.4. We highlight several limitations in existing evaluation practices. First, prior studies [185, 187, 189, 191] often compute MAUVE with relatively weak embedding models (e.g., all-MiniLM-L6-v2 or sentence-T5), which can inflate scores and mask quality differences. Second, these evaluations typically use short context lengths (e.g., 128 or 256 tokens), truncating longer texts and failing to capture overall generation quality. We address both issues by using stronger, domain-specific embeddings and a longer context

window. A more detailed discussion is provided in Appendix D.4.1.

Experimental results Comparison with baselines. We begin by comparing the end-to-end performance of our conditional generation approaches with the baselines. For $\mathcal{S}_2$, we use DP-FT for both the feature and conditional generator, while for $\mathcal{S}_3$ we adopt the optimal configuration ACTG, which we will discuss shortly in the ablation studies. As shown in Fig. 24 (rows 1 and 2), our methods consistently outperform Aug-PE, vanilla DP-FT and CTCL across all datasets, privacy levels, and evaluation metrics. These results persist when scaling to a larger base model (`gemma-3-4b-pt`; Appendix D.4.4) and when replacing the proprietary oracle with an open-source alternative (`Qwen2.5-32B-Instruct`; Appendix D.4.5). Together, these findings provide strong empirical support for the core hypothesis of our framework: *decoupling feature learning and conditional text generation yields a superior privacy-utility trade-off.* **Comparison of feature design $\mathcal{S}$.** We next compare the three feature designs within our framework. Results show that the rich tabular schema ($\mathcal{S}_3$) performs best, followed by the free-form summary ($\mathcal{S}_2$), both substantially outperforming the topic model ($\mathcal{S}_1$) used in CTCL. Unlike $\mathcal{S}_3$ and $\mathcal{S}_2$, which are tailored to each private dataset, the generic topic model in CTCL can suffer from domain mismatch; this underscores the importance of domain-specific features. Moreover, the superiority of the tabular schema $\mathcal{S}_3$ over free-form text $\mathcal{S}_2$ highlights the value of a compact yet informative schema that captures key information about the private dataset with minimal bits. This is further supported by our investigations on the impact of the semantic richness of $\mathcal{S}_3$ (Appendix D.4.3). These observations resonate with the notion of *compact representation* discussed in Hu et al. [33].

Ablation studies. We focus on $\mathcal{S}_3$ and conduct a modular evaluation to dissect errors at different stages. Fig. 25 illustrates three sources of error: **extraction error** ($d_{\mathrm{JS}}^{f_0}$) from LLM-based annotations of private text; **feature learning error** ($d_{\mathrm{JS}}^{f_1}$) introduced in Stage 1; and **conditional generation error** ($d_{\mathrm{JS}}^{f_2}$) introduced in Stage 2. Measuring

extraction error serves to validate the reliability of LLM annotations, which form the basis of our evaluation, while analyzing feature learning and conditional generation errors enables us to identify optimal configurations for these stages.

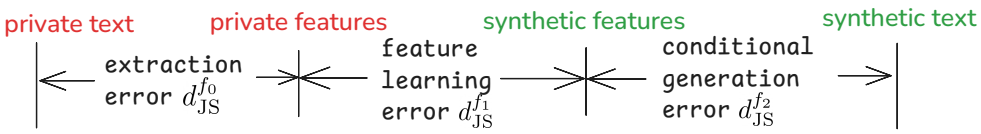


Figure 25: Fine-grained error analysis in our framework.

Evaluation approach. For $d_{\text{JS}}^{f_0}$, since ground-truth features of D_{priv}^{x} are unavailable, we perform five independent extractions and treat their *average distribution* as ground truth. We then compute the Jensen-Shannon distance between each trial and this average, and average across all attributes. For $d_{\text{JS}}^{f_1}$, we compute the distance between D_{priv}^{f} and $D_{\text{syn}}^{\tilde{f}}$. Finally, for $d_{\text{JS}}^{f_2}$, we compute the distance between $D_{\text{priv}}^{\tilde{f}}$ and the attributes extracted from $D_{\text{syn}}^{\tilde{x}}$.

Results. We present ablation results on bioRxiv in Fig. 24. In **Stage 0**, the extraction error $d_{\text{JS}}^{f_0}$ is around 0.01, confirming that LLM-extracted features are reliable. For **Stage 1**, we compare AIM and DP-FT as feature generators. Fig. 24(3a) shows that AIM achieves a much lower $d_{\text{JS}}^{f_1}$. One key reason is that AIM, as a specialized tabular synthesizer, allocates privacy budget only to predefined attributes of interest, rather than across all tokens as in DP-FT. This avoids wasting budget on non-sensitive information (e.g., JSON grammar) or public knowledge (e.g. age groups), thus improving the privacy-utility trade-off. Finally, Fig. 24(3b) shows that in **Stage 2**, DP-FT attains a lower $d_{\text{JS}}^{f_2}$ than direct prompting; we defer further discussion to Appendix D.4.7.

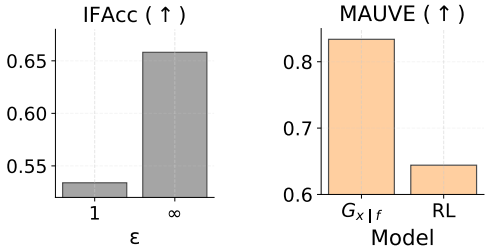
Taken together, these ablations empirically confirm our *optimal* configuration: a rich structured tabular schema ($\mathcal{S}_3$), AIM feature generator (G_f), and a DP-FT conditional generator ($G_{x|f}$). We refer to this configuration as **ACTG: Attribute-Conditioned Text Generation**, which establishes a new state of the art in DP synthetic text generation.

We also provide a comparative error analysis across the three stages of ACTG. As

shown in Fig. 24(3c), extraction error is negligible, while conditional generation incurs a larger error than feature learning, suggesting greater room for improvement in Stage 2. We will revisit this point in Sec. 6.3.3.

6.3 Boosting Fine-Grained Control in ACTG with Anchored RL

(a) DP degrades IFAcc **(b)** RL hurts MAUVE



(c) Example of reward hacking: TL;DR-style generation

Input feature: {"primary_research_area": "Neuroscience", "model_organism": "Drosophila melanogaster", "experimental_approach": "Wet Lab Experimentation", "dominant_data_type": "Phenotypic / Behavioral", "research_focus_scale": "Cellular", "disease_mention": "No Specific Disease Mentioned", "sample_size": "Relies on Cell/Animal Replicates", "research_goal": "Investigating a mechanism"}

Generated abstract: We experimentally evaluated whether larval synaptic plasticity is preserved by modulating spatial memory formation in *Drosophila*.

Figure 26: **(a)** IFAcc of the conditional generator $G_{x|f}$ with and without DP, showing a substantial drop under DP. **(b)** MAUVE score of generated text after RL, demonstrating a sharp decline in textual fidelity. **(c)** Example generation from the bioRxiv dataset that perfectly satisfies the input requirement (score: 8/8; see Appendix D.4.13) but fails to match the target domain (paper abstract). This occurs during RL training, where the model exploits the rubric reward and exhibits reward hacking.

So far, we have developed a general framework and identified its optimal configuration, ACTG, which produces high-quality DP synthetic *datasets*. However, static datasets are only part of the story. In practice, users may also require *controlled, on-demand generation* that satisfies specific requirements (e.g., an email with a positive tone on a given topic). In this setting, the focus shifts from *aggregate* dataset-level metrics like fidelity and utility to the generator's *per-instance* ability to reliably follow instructions.

In this section, we study the instruction-following capability of ACTG's conditional generator $G_{x|f}$ and show that it is significantly degraded under DP. To address this challenge, we propose Anchored RL, a post-training method built upon ACTG that strengthens control while preserving alignment with D^x_{priv}. Importantly, using RL to enhance control is made

possible by the conditional generation framework in Sec. 6.2 and ACTG's design, where tabular features serve as explicit, verifiable rewards.

6.3.1 Measuring and Improving Instruction Following

We focus on the structured tabular schema ($\mathcal{S}_3$, with K fields) used in ACTG and introduce the metric of **instruction following accuracy** (IFAcc). For a given input f, the generator $G_{x|f}$ produces a text x, from which a feature $\hat{f}$ is extracted using M_{oracle}. The *per-instance* IFAcc is defined as the fraction of fields in $\hat{f}$ that correctly match the input instruction f, and the overall IFAcc is the average of these per-instance scores across all text features. Formally:

$$\text{IFAcc} := \mathbb{E}_{f \sim D^f_{\text{priv}}} \left[\frac{1}{K} \sum_{k=1}^{K} \mathbb{I}(f_k = \hat{f}_k) \right], \tag{1}$$

where $\mathbb{I}(\cdot)$ is the indicator function and the expectation is taken over the private feature set.

Evaluating the conditional generator $G_{x|f}$ in ACTG, we find its instruction-following accuracy is significantly degraded by DP (e.g., from 66% to 53% on bioRxiv; see Fig. 26(a)). This loss of fine-grained control, even when aggregate metrics remain high, motivates a post-training procedure to restore instruction-following in conditional generation.

Boosting control via RL. Because ACTG is built on tabular features, it provides a natural interface for reinforcement learning. Each input feature f serves as a *rubric* for scoring generations. For each generated text x, we compute the per-instance IFAcc as the reward. The training loop is straightforward: we sample prompts from the DP feature generator ($f \sim G_f$), generate text ($x \sim G_{x|f}$), and use the resulting reward to update the model. We term this approach built on top of (and enabled by) ACTG as **ACTG-RL**. Crucially, unlike Wu et al. [140] who privatize the policy gradients, our RL training phase requires no additional privacy budget as both the prompts and the reward signal are derived without accessing the private data.

The reward hacking issue. ACTG-RL adopts the standard PPO objective [206], which reveals a clear trade-off between control and fidelity. While instruction following accuracy (IFAcc) improves, the MAUVE score plummets (see Fig. 26(b)), indicating a significant loss of textual quality. A closer examination of the outputs reveals the failure mode: the model learns to *hack* the reward by generating short "TL;DR"-style sentences. These outputs satisfy the rubric but fail to match the target domain's style (e.g., a full abstract); see Fig. 26(c). This failure highlights the need for a method that can boost control without sacrificing textual alignment. We address the challenge in what follows.

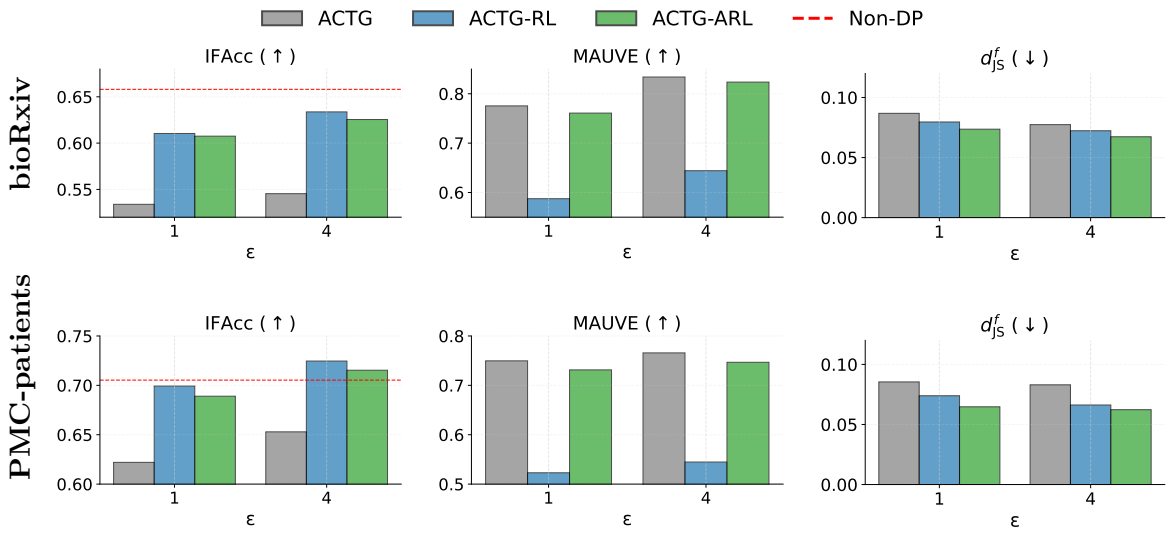


Figure 27: **Performance of the conditional generators before and after RL evaluated on three metrics.** ACTG-RL improves IFAcc but suffers from reward hacking, which collapses textual fidelity (MAUVE). ACTG-ARL resolves this trade-off, boosting IFAcc close to the non-DP level while maintaining high MAUVE and achieving the best attribute distribution matching.

6.3.2 A Post-Training Recipe: Anchored RL

We introduce Anchored RL (ARL), a post-training recipe for boosting instruction-following while preserving alignment with the original text distribution. It features two key design choices: a *hybrid training objective* to balance control and alignment, and a method for curating a high-quality *synthetic anchor dataset* without additional privacy cost.

Hybrid objective. Inspired by standard practices in RLHF [11, 12], our core idea is to mitigate reward hacking by anchoring the model to a reference distribution using a supervised fine-tuning (SFT) loss. The training objective thus becomes a hybrid of the standard RL loss and this SFT loss. However, this raises a critical question: what data should be used for the SFT anchor? Using the original private data would incur additional privacy cost, while the synthetic data sampled from $G_{x|f}$ suffers from the control issues that we aim to fix.

High-quality anchor via best-of-N sampling. We resolve this by crafting a high-quality, private dataset for the SFT objective using *best-of-N sampling*, a technique widely used in LLM alignment [207, 208]. For each feature f sampled from the DP generator G_f, we generate N candidate texts from $G_{x|f}$ and select the one with the highest per-instance IFAcc score. This uses additional test-time compute to distill a cleaner dataset, D_{SFT}, yielding a strong SFT anchor at no extra privacy cost. We further analyze the quality of the best-of-N dataset and the variance of per-instance IFAcc in Appendix D.4.14. **Putting together: Anchored RL.** Our final training recipe, ARL, combines these two components. We fine-tune from the DP-FT checkpoint $G_{x|f}$ using a hybrid objective that mixes the PPO gradient with the SFT gradient on curated best-of-N data: $\mathcal{L} = \mathcal{L}_{\text{RL}} + \gamma \cdot \mathcal{L}_{\text{SFT}}$. We employ a *linear decay* schedule for the coefficient γ, starting high to preserve text fidelity and gradually decreasing to allow for steady improvement in instruction following. This approach anchors the model, preventing it from drifting away from the desired text distribution while optimizing for control.

The end-to-end algorithm. Our final algorithm, **ACTG-ARL**, integrates ACTG with ARL into a single cohesive pipeline. It consists of four stages: private feature extraction, training the initial DP generators, curating the anchor dataset, and performing ARL training. The outputs are a DP synthetic dataset and a DP conditional generator with strong instruction-following capabilities. A detailed description is provided in Alg. 6 of Appendix D.2.

6.3.3 Experiments

We evaluate three models: 1) ACTG, the conditional generator $G_{x|f}$ from Sec. 6.2 which serves as the baseline; 2) ACTG-RL, where $G_{x|f}$ is further trained with a standard PPO objective; and 3) ACTG-ARL, where $G_{x|f}$ is trained with the hybrid objective and the best-of-N anchor dataset D_{SFT_N}, corresponding to our Anchored RL recipe. In Appendix D.4.12, we also consider a variant of ACTG where the base PT model is replaced with an IT model for DP-FT. In Appendix D.4.15, we showcase the importance of D_{SFT_N} and RL via ablation studies. Details of experimental setups are in Appendix D.3.4.

Results. Fig. 27 highlights the trade-off between control and fidelity. Baseline ACTG achieves strong fidelity (high MAUVE) but weak control (low IFAcc), while ACTG-RL improves IFAcc at the cost of severe reward hacking and a collapse in MAUVE. In contrast, ACTG-ARL resolves this tension, matching ACTG-RL in IFAcc while retaining ACTG's high fidelity. Importantly, this improvement in *per-instance* control also reduces the *end-to-end* error, as evidenced by the lowest d^f_{JS}. Since d^f_{JS} is a *metric* [209], the triangle inequality implies $d^f_{\mathrm{JS}} \leq d^{f_1}_{\mathrm{JS}} + d^{f_2}_{\mathrm{JS}}$, explaining why advances in Stage 2 reduce overall error, echoing our discussion in Sec. 6.2.3. We additionally evaluate the utility of synthetic data produced by ACTG-ARL in Appendix D.4.16.

6.4 Related Work

DP synthetic text. Research on DP synthetic text generation broadly falls into two paradigms: DP fine-tuning (DP-FT) [185, 210, 211, 212, 213] and Private Evolution (PE) [187, 188, 190]. Fine-tuning-based approaches learn the private text distribution implicitly via next-token prediction, while PE-based approaches leverage the power of LLMs to create a large pool of samples and iteratively refine them using embedding-space similarity. A recent trend in fine-tuning-based approaches is to improve data quality through *distribution alignment*. This can be done *post-hoc*, for instance by filtering synthetic data [189], or *a priori*, by using topics to condition the generation process [191]. Our work

builds upon and significantly extends this conditional approach: we abstract the concept into a general hierarchical framework and enhance it with a novel post-training RL recipe to improve fine-grained control.

Conditional text generation. Existing approaches to conditional text generation often rely on low-level signals to steer a model's output. For example, Putta et al. [214] adjust a model's hidden states using feedback from an attribute classifier, while DeSalvo et al. [215] use soft prompts to guide generation. Although CTCL [191] employs conditioning, its reliance on a fixed, general-purpose topic model limits its applicability to specialized datasets: the model may suffer from distribution mismatch, and many predefined topics may be irrelevant, yielding zero counts. In contrast, our framework uses human-interpretable, domain-specific features, offering more flexible and transparent control over the generated text.

6.5 Conclusion

We introduced a hierarchical framework (with ACTG as the optimal configuration) and a novel Anchored RL recipe that, together, form our end-to-end algorithm **ACTG-ARL**. Our approach delivers: 1) state-of-the-art DP synthetic text datasets and 2) a controllable, instruction-following generator. More broadly, our work elevates *control* as a third critical dimension, alongside utility and privacy, in DP synthetic text generation, with substantial practical benefits.

Chapter 7 Conclusion and Future Work

This thesis has argued that advancing generative AI requires treating data not as a passive training input, but as a first-class object of study. We pursued this vision along two complementary directions, data attribution and data privacy, developing theory and methods that quantify the value of data, optimize its use, safeguard its privacy, and unlock its potential through synthetic generation. We conclude by summarizing the main contributions and discussing future directions.

7.1 Summary of Contributions

On the attribution side, we established a theoretical foundation for scalable influence functions by showing that effective dimension, rather than rank, governs the sketch size required for random projection to preserve influence (Chapter 3). We also extended data attribution to online reinforcement learning through a local framework that attributes checkpoint updates to individual records, yielding both interpretability and a practical filtering algorithm that improves training efficiency (Chapter 4).

On the privacy side, we showed that the same formal DP guarantee can conceal substantial variation in empirical privacy behavior, and identified hyperparameter choice as a critical and previously underexplored factor (Chapter 5). We further proposed a hierarchical framework for DP conditional text generation, together with an anchored RL post-training method that improves both generation quality and control (Chapter 6).

Across these contributions, a common theme emerges: sustained progress in generative AI depends not on treating data as a commodity to be consumed, but on understanding its structure, value, and risks.

7.2 Future Directions

We organize future directions into two categories: technical extensions of the work in this thesis, and broader research themes on the science of data in generative AI.

7.2.1 Technical Extensions

From approximation error to modeling bias in influence functions. Our theory studies how well projected influence approximates its unprojected counterpart. An important but still underexplored direction is to understand how projection, regularization, and curvature approximation affect the quality of influence functions themselves as estimators of the underlying leave-one-out (LOO) quantity. In particular, metrics such as LOO correlation or LDS [38] conflate two distinct effects: the *approximation error* introduced by projection, and the *modeling bias* induced by regularization and curvature approximation. Disentangling these effects, and characterizing how projection and related techniques jointly shape both, remains an important open problem.

Attribution for reasoning and agentic systems. An important direction is to extend our framework to online RL algorithms used in reasoning and agentic systems, especially those for LLM training such as GRPO [92, 216, 217]. At the technical level, the framework should generalize whenever the attribution target and the corresponding per-sample gradients are well defined. At the application level, attribution offers a principled alternative to largely heuristic data selection methods [218, 219, 220, 221], and may provide new tools for improving reasoning performance in these systems.

Beyond (ε, δ): interpreting and reporting DP in practice. As differential privacy is increasingly adopted in generative AI, an important open question is how its guarantees should be interpreted and reported in practice. Our results suggest that reporting only (ε, δ) may obscure important aspects of privacy behavior, motivating both deeper scientific understanding and more informative reporting standards. Promising directions include using data attribution [29] and mechanistic interpretability [222] to study what DP does and does not promise in generative models, as well as developing reporting protocols such as GDP [223] that better capture practically relevant factors in DP training.

Attribution meets privacy. A natural but largely unexplored question lies at the intersection of the two themes of this thesis: what happens when data attribution methods are applied to models trained with differential privacy? DP-SGD is designed to limit the influence of any single training example, which suggests that attribution scores should be suppressed or homogenized under strong privacy guarantees. Whether this suppression is uniform across examples or selective—and how it interacts with the empirical privacy variance studied in Chapter 5—remains an open question. More broadly, attribution methods could serve as empirical probes for the effectiveness of privacy mechanisms, complementing formal guarantees with a data-level view of what DP training actually achieves. Conversely, understanding how privacy constraints reshape the landscape of data influence may lead to more faithful attribution methods that account for the training procedure rather than treating the model as a black box.

Private synthetic data beyond DP fine-tuning. Our hierarchical framework for DP text generation separates structured feature learning from conditional generation, and this modular design may extend beyond fine-tuning-based methods. A concrete next step is to investigate whether metadata-conditioned generation can improve non-fine-tuning approaches such as private evolution [187, 197].

7.2.2 Broader Themes

From post-hoc attribution to proactive data acquisition. Classical data attribution is typically formulated around a single fixed empirical risk minimization problem, static training and validation sets, and access to a trained model whose behavior is analyzed post hoc. This framing is often too narrow for generative AI, where models are developed through multiple stages of training, evaluation targets evolve over time, and the cost of training makes it important to identify valuable data before training takes place. While attribution in its classical form remains important, for example for fairly compensating

data contributors, the future of generative AI calls for a broader view: new paradigms for identifying data value, adaptive evaluation protocols that reflect the dynamic development cycle of generative models, and more proactive approaches to data acquisition and selection.

From formal privacy to pragmatic privacy. As generative AI systems become increasingly interactive, persistent, and agentic, privacy can no longer be understood solely through idealized threat models tied to a single training procedure. Modern systems create new surfaces for privacy risk through memory, tool use, long-horizon interaction, and deployment-time data flows, while formal guarantees such as differential privacy remain indispensable but necessarily incomplete: they rigorously bound one class of leakage, but do not by themselves characterize all privacy risks that arise in end-to-end systems. A broader agenda for the science of data is therefore to make privacy more practical and decision-relevant: to understand precisely what formal guarantees do and do not protect, to evaluate privacy at the level of complete systems rather than isolated models, and to treat privacy as a multi-faceted property that must be measured, interpreted, and managed in context rather than reduced to a single number.

From heuristic synthesis to feedback-driven data generation. As web-scale corpora approach saturation, continued progress in generative AI will increasingly depend on creating new data that expands model capabilities beyond what naturally occurring datasets can provide. Synthetic data is a natural candidate, but current pipelines remain largely heuristic and often lack reliable mechanisms for evaluating and improving the data they generate. A promising direction is to close this loop by using attribution signals to guide data generation itself: filtering low-quality synthetic samples, identifying capability gaps that call for targeted synthesis, and tracking how the value of synthetic data evolves as models improve. To be effective, such signals must remain adaptive and anchored to real-world objectives rather than overfitting to a fixed evaluation set.

A Appendix for Chapter 3: Theory of Random Projection

A.1 Proofs for Sec. 3.1.1 (Unregularized Projection)

In this section, we prove Theorem 3.1, which we first repeat the statement for convenience:

Theorem. The equality $\tau_0(g, g') = \tilde{\tau}_0(g, g')$ holds for any $g, g' \in \text{range}(F)$ *iff* P is injective on $\text{range}(F)$, i.e. $\text{rank}(PU) = \text{rank}(F) = r$ where $F = U\Lambda U^\top$ is the compact eigendecomposition of F with $U \in \mathbb{R}^{d\times r}$ orthonormal and $\Lambda \in \mathbb{R}^{r\times r}$ positive definite. Subsequently, for any PSD $F \in \mathbb{R}^{d\times d}$ and **any** matrix $P \in \mathbb{R}^{m\times d}$, one cannot hope to obtain any multiplicative approximation of $\tau_0(g, g')$ via $\tilde{\tau}_0(g, g')$ when $\text{rank}(PU) < r$.

Proof. For the "if" direction, suppose $\text{rank}(PU) = r$. Let $A := PU\Lambda^{1/2} \in \mathbb{R}^{m\times r}$ and it follows that A has full column rank. Then for any $g \in \text{range}(U) = \text{range}(F)$, write $g = Uz$ and $g'Uz'$ for some $z, z' \in \mathbb{R}^r$ and note $Pg = PUz = A\Lambda^{-1/2}z$ and similarly, $Pg' = A\Lambda^{-1/2}z'$, and $PFP^\top = AA^\top$. For full-column-rank A, $A^\top(AA^\top)^\dagger A = I_r$. Therefore

$$(Pg)^\top(PFP^\top)^\dagger(Pg') = z^\top\Lambda^{-1/2}A^\top(AA^\top)^\dagger A\Lambda^{-1/2}z' = z^\top\Lambda^{-1}z' = g^\top F^\dagger g'.$$

For the "only if" direction, suppose $\text{rank}(PU) < r$. Then there exists a nonzero $z \in \mathbb{R}^r$ such that $PUz = 0$. Let $g = Uz \in \text{range}(F)$ be the corresponding vector. Then, as $g^\top F^\dagger g = z^\top\Lambda^{-1}z > 0$, $(Pg)^\top(PFP^\top)^\dagger(Pg) = 0 \neq g^\top F^\dagger g > 0$, proving the result. □

A.2 Proofs for Sec. 3.1.2 (Regularized Projection)

This section collects technical results used in Sec. 3.1.2 that are omitted in the main text.

A.2.1 Proof of Resolvent Perturbation Concentration for Regularized Projection

We prove the key operator-norm perturbation step used in the proof of Theorem 3.2.[13] The general idea is to use the concentration of the sample covariance (Theorem 1.2) to control the resolvent-type map $A \mapsto A(A + \lambda I)^{-1}$ in operator norm, enabling the comparison of $F(F + \lambda I)^{-1}$ and $G(G + \lambda I)^{-1}$ in the proof of Theorem 3.2.

To prove Theorem 1.2, the key input is a standard high-probability covariance estimation bound for sub-Gaussian vectors (Vershynin [64, Exercise 9.2.5]), which we restate and prove as Theorem 1.1.

Proposition 1.1 (High-Probability Covariance Estimation)**.** *Let $\Sigma \succeq 0$ and let $X, X_1, \dots, X_m \in \mathbb{R}^d$ be i.i.d. mean-zero sub-Gaussian random vectors with covariance $\Sigma = \mathbb{E}[XX^\top]$. Define the sample covariance*

$$\Sigma_m := \frac{1}{m} \sum_{i=1}^{m} X_i X_i^\top.$$

Then for any $u \geq 0$, with probability at least $1 - 2e^{-u}$,

$$\|\Sigma_m - \Sigma\|_2 \leq C \left(\sqrt{\frac{r(\Sigma) + u}{m}} + \frac{r(\Sigma) + u}{m} \right) \|\Sigma\|_2,$$

where $r(\Sigma) := \operatorname{tr}(\Sigma)/\|\Sigma\|_2$ is the stable rank of $\Sigma^{1/2}$ and $C > 0$ is a universal constant.

Proof. Write $X = \Sigma^{1/2} Z$, where Z is an isotropic, mean-zero, sub-Gaussian random vector, and similarly $X_i = \Sigma^{1/2} Z_i$ with i.i.d. copies $Z_1, \dots, Z_m$. Let $A \in \mathbb{R}^{m \times d}$ be the matrix whose i-th row is $Z_i^\top$. As in the proof of Vershynin [64, Theorem 9.2.4], define $T := \Sigma^{1/2} S^{d-1}$ where S^{d-1} denotes the Euclidean unit sphere, then

$$\|\Sigma_m - \Sigma\|_2 = \frac{1}{m} \sup_{x \in T} \left| \|Ax\|_2^2 - m\|x\|_2^2 \right|.$$

[13] This can be viewed as a special case of approximate matrix multiplication for sub-Gaussian sketches; see Cohen et al. [224, Theorem 1]. Here, we state and prove the special case for clarity.

Consider the stochastic process

$$Y_x := \|Ax\|_2 - \sqrt{m}\|x\|_2, \qquad x \in T.$$

By Vershynin [64, Theorem 9.1.3], $(Y_x)_{x\in T}$ has sub-Gaussian increments. Applying the high-probability Talagrand comparison inequality [225, Theorem 3.2], we obtain that with probability at least $1 - 2e^{-v^2}$,

$$\sup_{x\in T}|Y_x| \le C\left(\gamma(T) + v\,\mathrm{rad}(T)\right),$$

where $\mathrm{rad}(T) := \sup_{x\in T}\|x\|_2$ denotes the *radius* of T, and $\gamma(T) := \mathbb{E}\big[\sup_{x\in T}|\langle g, x\rangle|\big]$ denotes the *Gaussian complexity* of T, for $g \sim \mathcal{N}(0, I_d)$.

Since $T = \Sigma^{1/2}S^{d-1}$, we have $\mathrm{rad}(T) = \|\Sigma\|_2^{1/2}$. Moreover,

$$\gamma(T) = \mathbb{E}\big[\|\Sigma^{1/2}g\|_2\big] \le \sqrt{\mathbb{E}[g^\top\Sigma g]} = \sqrt{\mathbb{E}[\mathrm{tr}(\Sigma g g^\top)]} = \sqrt{\mathrm{tr}(\Sigma)},$$

where the inequality follows from Jensen's inequality. Setting $u = v^2$ and recalling that $\mathrm{tr}(\Sigma) = r(\Sigma)\|\Sigma\|_2$, we conclude that, with probability at least $1 - 2e^{-u}$,

$$\sup_{x\in T}|Y_x| \le C\|\Sigma\|_2^{1/2}(\sqrt{r(\Sigma)} + \sqrt{u}).$$

Fix $x \in T$ and write $a := \|Ax\|_2$ and $b := \sqrt{m}\|x\|_2$. Then $b \le \sqrt{m}\|\Sigma\|_2^{1/2}$ and

$$|a^2 - b^2| \le |a - b|(|a - b| + 2b).$$

Using the bound above on $|a - b|$ and the fact that $b \ge 0$, we obtain

$$\sup_{x\in T}|a^2 - b^2| \le C\|\Sigma\|_2(\sqrt{r(\Sigma)} + \sqrt{u})\left(\sqrt{r(\Sigma)} + \sqrt{u} + \sqrt{m}\right).$$

Dividing by m yields

$$\|\Sigma_m - \Sigma\|_2 \le C\|\Sigma\|_2 \left(\frac{r(\Sigma) + u}{m} + \sqrt{\frac{r(\Sigma) + u}{m}} \right),$$

where we used $(\sqrt{r(\Sigma)} + \sqrt{u})^2 \lesssim r(\Sigma) + u$. This completes the proof. □

We now prove the concentration of sample covariance formally.

Lemma 1.2. *Let $P \in \mathbb{R}^{m \times d}$ be a sketching matrix whose rows are given by $P_i^\top = \frac{1}{\sqrt{m}} W_i^\top$, where $\{W_i\}_{i=1}^m \sim W$ are i.i.d. sub-Gaussian random vectors in $\mathbb{R}^d$ satisfying $\mathbb{E}[W] = 0$ and $\mathbb{E}[WW^\top] = I_d$. Let $M \in \mathbb{R}^{d \times s}$ be a matrix and define $\Sigma \coloneqq M^\top M$. For any $\varepsilon, \delta \in (0,1)$, if*

$$m = \Omega\left(\frac{r(\Sigma) + \log(1/\delta)}{\varepsilon^2} \right),$$

where $r(\Sigma) = \operatorname{tr}(\Sigma)/\|\Sigma\|_2$ is the stable rank of $\Sigma^{1/2}$, then with probability at least $1 - \delta$,

$$\|M^\top (P^\top P - I_d) M\|_2 \le \varepsilon \|M\|_2^2.$$

Proof. The rows of P satisfy $P_i^\top = \frac{1}{\sqrt{m}} X_i^\top$, where $\{X_i\}_{i=1}^m$ are i.i.d. isotropic sub-Gaussian vectors. Observe that

$$M^\top P^\top P M = M^\top \left(\frac{1}{m} \sum_{i=1}^m X_i X_i^\top \right) M = \frac{1}{m} \sum_{i=1}^m (M^\top X_i)(M^\top X_i)^\top =: \Sigma_m.$$

Define $Y_i \coloneqq M^\top X_i$. Then $\{Y_i\}_{i=1}^m$ are i.i.d. mean-zero sub-Gaussian vectors with covariance

$$\mathbb{E}[YY^\top] = M^\top \mathbb{E}[XX^\top] M = M^\top M = \Sigma.$$

Applying Vershynin [64, Exercise 9.2.5, Theorem 1.1] yields that, with probability at least $1 - 2e^{-u}$,

$$\|\Sigma_m - \Sigma\|_2 \le C \left(\sqrt{\frac{r(\Sigma) + u}{m}} + \frac{r(\Sigma) + u}{m} \right) \|\Sigma\|_2,$$

Choosing $m \geq (r(\Sigma) + u)/\varepsilon^2$ ensures $\sqrt{(r(\Sigma) + u)/m} \leq \varepsilon$ and $(r(\Sigma) + u)/m \leq \varepsilon^2 < \varepsilon$ for $\varepsilon < 1$. Since $\|\Sigma\|_2 = \|M^\top M\|_2 = \|M\|_2^2$, we conclude that

$$\|M^\top P^\top P M - M^\top M\|_2 = \|M^\top (P^\top P - I_d) M\|_2 \leq \varepsilon \|M\|_2^2.$$

Setting $u = \Theta(\log(1/\delta))$ completes the proof. □

We can now state and prove the concentration of resolvent perturbation for regularized projection as follows:

Lemma 1.3. *Let $F \succeq 0$ and $\lambda > 0$, and define $G = F^{1/2} P^\top P F^{1/2}$. Then for any $\varepsilon, \delta \in (0, 1)$, if $m = \Omega(\varepsilon^{-2}(d_\lambda(F) + \log(1/\delta)))$, with probability at least $1 - \delta$,*

$$\|F(F + \lambda I)^{-1} - G(G + \lambda I)^{-1}\|_2 \leq \varepsilon.$$

Proof. Applying Theorem 1.2 with $M = B = F^{1/2}(F + \lambda I)^{-1/2}$, for any $\delta, \epsilon > 0$, if $m = \Omega(\epsilon^{-2}(r(B^\top B) + \log(1/\delta)))$ then with probability at least $1 - \delta$,

$$\|B^\top (P^\top P - I_d) B\|_2 \leq \epsilon \|B\|_2^2.$$

We first note that if $\|B\|_2^2 = 0$, then the bound is trivial. Assuming $\|B\|_2 > 0$. Then we see that $\|B\|_2^2 = \|B^\top B\|_2 = \|F(F + \lambda I)^{-1}\|_2 \leq 1$ since the eigenvalues of $F(F + \lambda I)^{-1}$ equal $\lambda_i(F)/(\lambda_i(F) + \lambda)$. Now, pick $\epsilon := \min(1, \varepsilon/2\|B\|_2)$, and note that $\|B\|_F^2 = \operatorname{tr}(F(F + \lambda I)^{-1}) = d_\lambda(F)$, we have

$$r(B^\top B) = \frac{\operatorname{tr}(B^\top B)}{\|B^\top B\|_2} = \frac{\|B\|_F^2}{\|B\|_2^2} = \frac{d_\lambda(F)}{\|B\|_2^2}.$$

After substitution, with $\|B\|_2^2 \leq 1$, we conclude that if

$$m = \Omega\left(\epsilon^{-2}\left(\frac{d_\lambda(F)}{\|B\|_2^2} + \log(1/\delta)\right)\right) = \Omega\left(\varepsilon^{-2}\left(d_\lambda(F) + \log(1/\delta)\right)\right),$$

we have $\|B^\top(P^\top P - I_d)B\|_2 \le \epsilon\|B\|_2^2 \le \varepsilon/2$. This implies

$$-\frac{\varepsilon}{2}I \preceq B^\top(P^\top P - I)B \preceq \frac{\varepsilon}{2}I \implies B^\top B - \frac{\varepsilon}{2}I \preceq B^\top P^\top PB \preceq B^\top B + \frac{\varepsilon}{2}I.$$

With

$$B^\top B = (F+\lambda I)^{-1/2}F(F+\lambda I)^{-1/2},$$
$$B^\top P^\top PB = (F+\lambda I)^{-1/2}\underbrace{F^{1/2}P^\top PF^{1/2}}_{G}(F+\lambda I)^{-1/2},$$

we can conjugate by $(F+\lambda I)^{1/2}$, which yields

$$\left(1-\frac{\varepsilon}{2}\right)F - \frac{\varepsilon}{2}\lambda I \preceq G \preceq \left(1+\frac{\varepsilon}{2}\right)F + \frac{\varepsilon}{2}\lambda I.$$

Adding λI gives

$$\left(1-\frac{\varepsilon}{2}\right)(F+\lambda I) \preceq G + \lambda I \preceq \left(1+\frac{\varepsilon}{2}\right)(F+\lambda I).$$

Define $S \coloneqq (F+\lambda I)^{-1/2}(G+\lambda I)(F+\lambda I)^{-1/2}$. Conjugating the above by $(F+\lambda I)^{-1/2}$ yields

$$\left(1-\frac{\varepsilon}{2}\right)I \preceq S \preceq \left(1+\frac{\varepsilon}{2}\right)I.$$

Hence, $S \succ 0$ and $\|S-I\|_2 \le \varepsilon/2$ and $\|S^{-1}\|_2 \le \frac{1}{1-\varepsilon/2}$. From the definition of S,

$$(G+\lambda I)^{-1} = (F+\lambda I)^{-1/2}S^{-1}(F+\lambda I)^{-1/2},$$

hence

$$(G+\lambda I)^{-1} - (F+\lambda I)^{-1} = (F+\lambda I)^{-1/2}(S^{-1}-I)(F+\lambda I)^{-1/2},$$

giving

$$\|(G+\lambda I)^{-1}-(F+\lambda I)^{-1}\|_2 \le \|(F+\lambda I)^{-1}\|_2 \|S^{-1}-I\|_2.$$

From the identity $S^{-1}-I = S^{-1}(I-S)$, we have

$$\|S^{-1}-I\|_2 \le \|S^{-1}\|_2 \|S-I\|_2 \le \frac{\varepsilon/2}{1-\varepsilon/2}.$$

With $\|(F+\lambda I)^{-1}\|_2 \le 1/\lambda$, we have

$$\|(G+\lambda I)^{-1}-(F+\lambda I)^{-1}\|_2 \le \frac{1}{\lambda}\frac{\varepsilon/2}{1-\varepsilon/2}.$$

From the identity $A(A+\lambda I)^{-1} = I - \lambda(A+\lambda I)^{-1}$ for any PSD A, we have

$$F(F+\lambda I)^{-1} - G(G+\lambda I)^{-1} = \lambda\left((G+\lambda I)^{-1}-(F+\lambda I)^{-1}\right),$$

and hence

$$\|F(F+\lambda I)^{-1} - G(G+\lambda I)^{-1}\|_2 \le \lambda \frac{1}{\lambda}\frac{\varepsilon/2}{1-\varepsilon/2} = \frac{\varepsilon/2}{1-\varepsilon/2}.$$

Finally, note that $\frac{\varepsilon/2}{1-\varepsilon/2} \le \varepsilon$ for any $\varepsilon \in (0,1)$, this proves the result. □

A.2.2 OSE-Based Alternative Analysis

We record a self-contained proof of the OSE-based alternative analysis sketched in discussion following Theorem 3.2. Let $A \in \mathbb{R}^{d\times r}$ be a fixed matrix. A random matrix $P \in \mathbb{R}^{m\times d}$ is an *oblivious subspace embedding (OSE)* for range(A) with distortion $\varepsilon \in (0,1)$ if, with high probability,

$$(1-\varepsilon)\|Ax\|_2^2 \le \|PAx\|_2^2 \le (1+\varepsilon)\|Ax\|_2^2, \qquad \forall x \in \mathbb{R}^r.$$

Equivalently,

$$-\varepsilon A^\top A \preceq A^\top(P^\top P - I_d)A \preceq \varepsilon A^\top A.$$

It is well known that standard oblivious sketches (Gaussian, Rademacher, SJLT) satisfy this property provided $m = \Omega(\varepsilon^{-2} \operatorname{rank}(A))$ [67, Theorems 2.3 and 6.10]. In our case, we apply the OSE framework with $A = F^{1/2}$. It is straightforward to see that $\operatorname{rank}(A) = \operatorname{rank}(F) = r$, so achieving an ε-OSE for $\operatorname{range}(A)$ requires $m = \Omega(r/\varepsilon^2)$.

Define $G \coloneqq F^{1/2} P^\top P F^{1/2}$. The OSE condition gives

$$(1-\varepsilon)F \preceq G \preceq (1+\varepsilon)F.$$

Consider $f(t) \coloneqq \frac{t}{t+\lambda}$ for $t \geq 0$. Since $t \mapsto (t+\lambda)^{-1}$ is operator monotone decreasing on $[0, \infty)$, it follows that $f(t) = 1 - \lambda(t+\lambda)^{-1}$ is operator monotone increasing.

Applying f to the sandwich gives

$$f\big((1-\varepsilon)F\big) \preceq f(G) \preceq f\big((1+\varepsilon)F\big),$$

or

$$(1-\varepsilon)F\big((1-\varepsilon)F + \lambda I\big)^{-1} \preceq G(G+\lambda I)^{-1} \preceq (1+\varepsilon)F\big((1+\varepsilon)F + \lambda I\big)^{-1}.$$

Since F commutes with any function of itself, the resulting operator-norm deviation reduces to a scalar supremum. For example,

$$\big\| f\big((1+\varepsilon)F\big) - f(F) \big\|_2 = \sup_{t \geq 0} \left| \frac{(1+\varepsilon)t}{(1+\varepsilon)t + \lambda} - \frac{t}{t+\lambda} \right| = \sup_{t \geq 0} \frac{\varepsilon \lambda t}{\big((1+\varepsilon)t + \lambda\big)(t+\lambda)}.$$

The same bound holds with $(1+\varepsilon)$ replaced by $(1-\varepsilon)$. A short calculus argument shows the supremum is at most ε; hence

$$\big\| f(G) - f(F) \big\|_2 = \big\| G(G+\lambda I)^{-1} - F(F+\lambda I)^{-1} \big\|_2 \leq O(\varepsilon).$$

Combining the above operator control with the argument in the proof of Theorem 3.2 yields the same bilinear and quadratic influence error bounds. The key difference is the sample

complexity: the OSE route fundamentally scales with r, whereas our main analysis scales with the effective dimension $d_\lambda(F)$.

A.2.3 Proof of Anti-Concentration of Gaussian Sample Covariance

Next, we prove the worst-case lower bound (Theorem 3.4). The proof consists of two main components:

1. An anti-concentration result for the sample covariance of Gaussian matrices, which shows that deviations of order $\sqrt{k/m}$ occur with constant probability (Theorem 1.4).
2. A carefully constructed hard instance F for which such deviations translate directly into a large error in the regularized quadratic form.

We note that since the proof of Theorem 1.4 contains many technical computation, we defer them for a cleaner presentation after the main proof.

Lemma 1.4. *Let $W \in \mathbb{R}^{m\times k}$ have rows $w_1, \dots, w_m \sim \mathcal{N}(0, I_k)$ i.i.d., and define $S := \frac{1}{m}W^\top W$. Then for all $m, k \geq 1$,*

$$\Pr\left(\|S - I_k\|_2 \geq \frac{1}{2}\sqrt{\frac{k}{m}}\right) \geq \frac{3}{80}.$$

Proof. Define

$$A := S - I_k = \frac{1}{m}\sum_{i=1}^m X_i, \qquad X_i := w_i w_i^\top - I_k.$$

Then $\mathbb{E}[X_i] = 0$ and $X_1, \dots, X_m$ are independent. Let $g := \|A\|_F^2 \geq 0$. Expanding, we have

$$g = \left\|\frac{1}{m}\sum_{i=1}^m X_i\right\|_F^2 = \frac{1}{m^2}\sum_{i,j=1}^m \langle X_i, X_j\rangle.$$

Since $\mathbb{E}[\langle X_i, X_j\rangle] = 0$ for $i \neq j$ from independence and $\mathbb{E}[X_i] = 0$,

$$\mathbb{E}[g] = \frac{1}{m^2}\sum_{i=1}^m \mathbb{E}[\|X_i\|_F^2] = \frac{1}{m}\mathbb{E}[\|X_1\|_F^2].$$

A direct computation gives $\mathbb{E}[\|X_1\|_F^2] = k(k+1)$, hence

$$\mathbb{E}[g] = \frac{k(k+1)}{m}.$$

On the other hand, as $g = \frac{1}{m^2}\sum_{i,j} Y_{ij}$ where $Y_{ij} := \langle X_i, X_j\rangle$, we have

$$\mathbb{E}[g^2] = \frac{1}{m^4}\sum_{i,j,p,q} \mathbb{E}[Y_{ij}Y_{pq}].$$

By independence and centering, only overlapping index patterns contribute, and one obtains

$$\mathbb{E}[g^2] = \frac{1}{m^4}\left(ma + m(m-1)\mu^2 + 2m(m-1)b\right),$$

where $\mu := \mathbb{E}[\|X_1\|_F^2]$, $a := \mathbb{E}[\|X_1\|_F^4]$, and $b := \mathbb{E}[\langle X, Y\rangle^2]$, and $X := ww^\top - I_k$, $Y := uu^\top - I_k$ with $w \perp u$ i.i.d. $\mathcal{N}(0, I_k)$. Moreover, a moment calculations yield

$$\mu = k(k+1), \qquad a = k^4 + 10k^3 + 25k^2 + 24k, \qquad b = 2k^2 + 2k.$$

Substituting these expressions into the above formula for $\mathbb{E}[g^2]$ and simplifying gives the explicit comparison

$$\frac{(\mathbb{E}[g])^2}{\mathbb{E}[g^2]} \geq \frac{1}{15},$$

uniformly for all $m, k \geq 1$. Equivalently, $\mathbb{E}[g^2] \leq 15(\mathbb{E}[g])^2$. Then, by Paley–Zygmund [226], for any $\theta \in (0,1)$,

$$\Pr(g \geq \theta\mathbb{E}[g]) \geq (1-\theta)^2\frac{(\mathbb{E}[g])^2}{\mathbb{E}[g^2]} \geq \frac{(1-\theta)^2}{15}.$$

Taking $\theta = 1/4$ yields

$$\Pr\left(g \geq \frac{1}{4}\mathbb{E}[g]\right) \geq \frac{(3/4)^2}{15} = \frac{3}{80}.$$

On this event,

$$g = \|A\|_F^2 \geq \frac{1}{4}\cdot\frac{k(k+1)}{m}.$$

Using $\|A\|_F^2 \le k\|A\|_2^2$, we obtain

$$\|A\|_2^2 \ge \frac{1}{k}\|A\|_F^2 \ge \frac{1}{k}\cdot\frac{1}{4}\cdot\frac{k(k+1)}{m} = \frac{k+1}{4m} \ge \frac{k}{4m}.$$

Hence, with probability at least $3/80$,

$$\|A\|_2 = \|S - I_k\|_2 \ge \frac{1}{2}\sqrt{\frac{k}{m}}.$$

□

We now provide the routine calculations used in the proof of Theorem 1.4. In particular, we compute moments of Gaussian rank-one matrices and enumerate the index patterns in $\mathbb{E}[\|A\|_F^4]$.

Chi-square moments. Let $r \sim \chi_k^2$. For any $n \in \mathbb{N}_+$,

$$\mathbb{E}[r^n] = \prod_{i=0}^{n-1}(k+2i).$$

In particular,

$$\mathbb{E}[r] = k, \quad \mathbb{E}[r^2] = k(k+2), \quad \mathbb{E}[r^3] = k(k+2)(k+4), \quad \mathbb{E}[r^4] = k(k+2)(k+4)(k+6).$$

Moments of $X = ww^\top - I_k$. Let $w \sim \mathcal{N}(0, I_k)$ and define $X := ww^\top - I_k$. Write $r := \|w\|_2^2 \sim \chi_k^2$. We compute $\mu := \mathbb{E}[\|X\|_F^2]$ and $a := \mathbb{E}[\|X\|_F^4]$. First,

$$\begin{aligned}\|X\|_F^2 &= \operatorname{tr}(X^\top X) = \operatorname{tr}(X^2) = \operatorname{tr}\left((ww^\top - I_k)^2\right) \\ &= \operatorname{tr}(ww^\top ww^\top) - 2\operatorname{tr}(ww^\top) + \operatorname{tr}(I_k).\end{aligned}$$

By trace cyclicity, $\operatorname{tr}(ww^\top ww^\top) = \operatorname{tr}(w(w^\top w)w^\top) = (w^\top w)\operatorname{tr}(ww^\top) = r^2$, while $\operatorname{tr}(ww^\top) = r$ and $\operatorname{tr}(I_k) = k$. Hence

$$\|X\|_F^2 = r^2 - 2r + k.$$

Taking expectation and using the moments above gives

$$\mu = \mathbb{E}[r^2 - 2r + k] = k(k+2) - 2k + k = k(k+1).$$

Moreover,

$$a = \mathbb{E}(r^2 - 2r + k)^2 = \mathbb{E}\big[r^4 - 4r^3 + (4+2k)r^2 - 4kr + k^2\big].$$

Substituting $\mathbb{E}[r], \dots, \mathbb{E}[r^4]$ yields

$$a = k^4 + 10k^3 + 25k^2 + 24k.$$

The mixed term $b = \mathbb{E}[\langle X, Y\rangle^2]$. Let $w, u \sim \mathcal{N}(0, I_k)$ be independent, and define $X := ww^\top - I_k$ and $Y := uu^\top - I_k$. Set $r := \|w\|_2^2$, $s := \|u\|_2^2$, and $t := w^\top u$. A direct expansion gives

$$\langle X, Y\rangle = \operatorname{tr}\left((ww^\top - I_k)(uu^\top - I_k)\right) = t^2 - r - s + k,$$

since $\operatorname{tr}(ww^\top uu^\top) = \operatorname{tr}(w(w^\top u)u^\top) = (w^\top u)^2 = t^2$. Therefore,

$$b = \mathbb{E}[(t^2 - r - s + k)^2] = \mathbb{E}[t^4] + \mathbb{E}[(r+s-k)^2] - 2\mathbb{E}\big[t^2(r+s-k)\big].$$

To evaluate these terms, write $t = \sum_{\ell=1}^k Z_\ell$ with $Z_\ell := w_\ell u_\ell$. Then $\mathbb{E}[Z_\ell] = 0$, $\mathbb{E}[Z_\ell^2] = 1$, and $\mathbb{E}[Z_\ell^4] = 9$, and hence

$$\mathbb{E}[t^4] = \sum_{\ell=1}^k \mathbb{E}[Z_\ell^4] + 6\sum_{1\le i<j\le k} \mathbb{E}[Z_i^2]\mathbb{E}[Z_j^2] = 9k + 6\binom{k}{2} = 3k^2 + 6k.$$

Next, since $r, s \sim \chi^2_k$ are independent, we have $\mathbb{E}[r] = \mathbb{E}[s] = k$ and $\mathrm{Var}[r] = \mathrm{Var}[s] = 2k$, so

$$\mathbb{E}[(r+s-k)^2] = \mathrm{Var}[r+s-k] + (\mathbb{E}[r+s-k])^2 = 4k + k^2.$$

Finally, conditioning on w gives $t \mid w \sim \mathcal{N}(0, \|w\|_2^2) = \mathcal{N}(0, r)$, so $\mathbb{E}[t^2 \mid w] = r$ and hence $\mathbb{E}[t^2] = \mathbb{E}[r] = k$. Moreover,

$$\mathbb{E}[t^2 r] = \mathbb{E}[r\mathbb{E}[t^2 \mid w]] = \mathbb{E}[r^2] = k(k+2).$$

By symmetry, $\mathbb{E}[t^2(r+s-k)] = 2\mathbb{E}[t^2 r] - k\mathbb{E}[t^2] = k^2 + 4k$, so altogether

$$b = (3k^2 + 6k) + (k^2 + 4k) - 2(k^2 + 4k) = 2k^2 + 2k.$$

Enumerating index patterns in $\mathbb{E}\|A\|_F^4$. Let $A = \frac{1}{m}\sum_{i=1}^m X_i$ with $X_i = w_i w_i^\top - I_k$ i.i.d. and mean-zero, and set $Z = \|A\|_F^2$. With $Y_{ij} := \langle X_i, X_j \rangle$, we have

$$Z = \frac{1}{m^2}\sum_{i,j=1}^m Y_{ij}, \qquad Z^2 = \frac{1}{m^4}\sum_{i,j,p,q=1}^m Y_{ij}Y_{pq}, \qquad \mathbb{E}[Z^2] = \frac{1}{m^4}\sum_{i,j,p,q}\mathbb{E}[Y_{ij}Y_{pq}].$$

The expectation $\mathbb{E}[Y_{ij}Y_{pq}]$ is zero unless $\{i,j\} \cap \{p,q\} \neq \emptyset$. Indeed, if $\{i,j\} \cap \{p,q\} = \emptyset$, then the two factors depend on disjoint sets of independent random variables. Moreover, for $i \neq j$, $\mathbb{E}[Y_{ij}] = \mathbb{E}[\langle X_i, X_j \rangle] = \langle \mathbb{E}[X_i], \mathbb{E}[X_j] \rangle = 0$, so such disjoint products vanish. The only contributing configurations are:

(T1) $(i,j) = (p,q)$, contributing $\mathbb{E}[Y_{ij}^2]$;

(T2) $(i,j) = (q,p)$, contributing $\mathbb{E}[Y_{ij}Y_{ji}] = \mathbb{E}[Y_{ij}^2]$ since $Y_{ij} = Y_{ji}$;

(T3) $i = j$ and $p = q$ with $i \neq p$, contributing $\mathbb{E}[Y_{ii}]\mathbb{E}[Y_{pp}] = \mu^2$.

Counting multiplicities, type (T1) gives $\sum_{i,j}\mathbb{E}[Y_{ij}^2] = ma + m(m-1)b$, where $a = \mathbb{E}[\|X_1\|_F^4]$ (since $Y_{11} = \langle X_1, X_1 \rangle = \|X_1\|_F^2$) and $b = \mathbb{E}[\langle X, Y \rangle^2]$ for independent copies X, Y. Type (T2)

contributes another $m(m-1)b$, and type (T3) contributes $m(m-1)\mu^2$. Hence

$$\mathbb{E}[Z^2] = \frac{1}{m^4}\Big(ma + m(m-1)\mu^2 + 2m(m-1)b\Big).$$

Using $\mu = k(k+1)$, $a = k^4 + 10k^3 + 25k^2 + 24k$, and $b = 2k^2 + 2k$, one checks that for all $m, k \geq 1$,

$$\mathbb{E}[Z^2] \leq \frac{15k^2(k+1)^2}{m^2}.$$

A.2.4 Proof of Worst-Case Lower Bound

We restate Theorem 3.4 below for convenience:

Theorem. Let $P \in \mathbb{R}^{m\times d}$ be a Gaussian oblivious sketch with rows i.i.d. $\mathcal{N}(0, I_d)$. There exists a family of matrices $F \in \mathbb{R}^{d\times d}$ such that if $m = o(d_\lambda(F)/\varepsilon^2)$, then with constant probability, there exists $g \in \text{range}(F)$ such that

$$|\tilde{\tau}_\lambda(g,g) - \tau_\lambda(g,g)| = \Omega(\varepsilon)\tau_0(g,g).$$

Proof. Fix integers $k \leq r = \text{rank}(F) \leq d$ and define

$$F = \text{diag}(\underbrace{\lambda, \dots, \lambda}_{k}, \underbrace{\eta\lambda, \dots, \eta\lambda}_{r-k}, \underbrace{0, \dots, 0}_{d-r}),$$

where $\eta > 0$ will be chosen sufficiently small (as a function of ε and fixed constants only). Then

$$d_\lambda(F) = \sum_{i=1}^{d} \frac{\lambda_i(F)}{\lambda_i(F)+\lambda} = \frac{k\lambda}{\lambda+\lambda} + \frac{(r-k)\eta\lambda}{\eta\lambda+\lambda} = \frac{k}{2} + \frac{\eta}{1+\eta}(r-k) = \Theta(k) \quad \text{for } \eta \ll 1.$$

Let $P = \frac{1}{\sqrt{m}}W$ where $W \in \mathbb{R}^{m\times d}$ has i.i.d. $\mathcal{N}(0,1)$ entries, and partition

$$P = (P_L, P_S, P_Z)$$

according to the blocks of F, i.e. $P_L \in \mathbb{R}^{m\times k}$, $P_S \in \mathbb{R}^{m\times(r-k)}$. Choose

$$g = F^{1/2}y, \qquad y = \begin{pmatrix} y_L \\ 0 \\ 0 \end{pmatrix}, \quad \|y_L\|_2 = 1,$$

for some y. We see that $g_L = \sqrt{\lambda}y_L$. Now, we see that

$$\tau_\lambda(g,g) = g^\top(F+\lambda I)^{-1}g = y^\top F^{1/2}(F+\lambda I)^{-1}F^{1/2}y = y_L^\top \frac{\lambda}{\lambda+\lambda} I_k y_L = \frac{1}{2},$$

and

$$\tau_0(g,g) = g^\top F^\dagger g = y^\top y = \|y_L\|_2^2 = 1.$$

On the other hand, the sketched quantity equals

$$\tilde{\tau}_\lambda(g,g) = g^\top P^\top (PFP^\top + \lambda I)^{-1} Pg.$$

Since $g = F^{1/2}y$ and $F = \lambda\,\mathrm{diag}(I_k, \eta I_{r-k}, 0)$, we have $Pg = \sqrt{\lambda}P_L y_L$ and

$$PFP^\top + \lambda I = \lambda(P_L P_L^\top + \eta P_S P_S^\top + I).$$

Therefore

$$\tilde{\tau}_\lambda(g,g) = y_L^\top P_L^\top \left(P_L P_L^\top + \eta P_S P_S^\top + I\right)^{-1} P_L y_L.$$

Decomposing the error, write

$$\left|\tilde{\tau}_\lambda(g,g) - \tau_\lambda(g,g)\right| \geq T_1 - T_2,$$

where

$$T_1 \coloneqq \left| y_L^\top P_L^\top (P_L P_L^\top + I)^{-1} P_L y_L - \frac{1}{2} \right|,$$
$$T_2 \coloneqq \left| y_L^\top P_L^\top \left[(P_L P_L^\top + I)^{-1} - (P_L P_L^\top + \eta P_S P_S^\top + I)^{-1} \right] P_L y_L \right|.$$

Lower-Bounding T_1. Let $M \coloneqq P_L^\top P_L \in \mathbb{R}^{k \times k}$. Using the push-through identity

$$P_L^\top (P_L P_L^\top + I)^{-1} P_L = M(M+I)^{-1},$$

we have

$$\left| y_L^\top P_L^\top (P_L P_L^\top + I)^{-1} P_L y_L - \frac{1}{2} \right| = \left| y_L^\top M(M+I)^{-1} y_L - \frac{1}{2} \right|.$$

Let $\lambda_1, \dots, \lambda_k$ be the eigenvalues of M and choose y_L to be a unit eigenvector corresponding to an eigenvalue $\lambda_\star$. Then

$$y_L^\top M(M+I)^{-1} y_L = \frac{\lambda_\star}{\lambda_\star + 1},$$

and hence

$$\left| y_L^\top M(M+I)^{-1} y_L - \frac{1}{2} \right| = \left| \frac{\lambda_\star}{\lambda_\star + 1} - \frac{1}{2} \right| = \left| \frac{\lambda_\star - 1}{2(\lambda_\star + 1)} \right| = \frac{|\lambda_\star - 1|}{2(\lambda_\star + 1)}.$$

Using $\lambda_\star + 1 \le |\lambda_\star - 1| + 2$, we obtain

$$\left| y_L^\top M(M+I)^{-1} y_L - \frac{1}{2} \right| \ge \frac{|\lambda_\star - 1|}{2|\lambda_\star - 1| + 4} \ge \min\left\{ \frac{|\lambda_\star - 1|}{8}, \frac{1}{4} \right\}.$$

Now observe that $\|M - I\|_2 = \max_i |\lambda_i - 1|$, so if $\|M - I\|_2 \ge t$, then there exists $\lambda_\star$ with $|\lambda_\star - 1| \ge t$ and the above choice of y_L yields

$$\left| y_L^\top M(M+I)^{-1} y_L - \frac{1}{2} \right| \ge \min\left\{ \frac{t}{8}, \frac{1}{4} \right\}.$$

Since $P_L = \frac{1}{\sqrt{m}} W_L$ with $W_L \in \mathbb{R}^{m \times k}$ i.i.d. Gaussian rows, we have

$$M = P_L^\top P_L = \frac{1}{m} W_L^\top W_L.$$

Applying Theorem 1.4 to W_L gives

$$\Pr\left(\|M - I_k\|_2 \geq \frac{1}{2}\sqrt{\frac{k}{m}} \right) \geq \frac{3}{80}.$$

On this event we may take $t = \frac{1}{2}\sqrt{k/m}$ above, giving

$$\left| y_L^\top M (M+I)^{-1} y_L - \frac{1}{2} \right| \geq \min\left\{ \frac{1}{16}\sqrt{\frac{k}{m}}, \frac{1}{4} \right\},$$

with probability at least $3/80$.

Upper-Bounding T_2. Let $A \coloneqq P_L P_L^\top + I \succ 0$ and $B \coloneqq A + \eta P_S P_S^\top \succ 0$. By Woodbury matrix identity,

$$B^{-1} = (A + \eta P_S P_S^\top)^{-1} = A^{-1} - A^{-1} P_S (\eta^{-1} I + P_S^\top A^{-1} P_S)^{-1} P_S^\top A^{-1}.$$

Hence

$$A^{-1} - B^{-1} = A^{-1} P_S (\eta^{-1} I + P_S^\top A^{-1} P_S)^{-1} P_S^\top A^{-1}.$$

Using $|y_L^\top (\cdot) y_L| \leq \|\cdot\|_2$, we obtain

$$T_2 \leq \left\| P_L^\top (A^{-1} - B^{-1}) P_L \right\|_2 = \left\| P_L^\top A^{-1} P_S \left(\eta^{-1} I + P_S^\top A^{-1} P_S \right)^{-1} P_S^\top A^{-1} P_L \right\|_2.$$

Define

$$X \coloneqq A^{-1/2} P_L, \qquad C \coloneqq A^{-1/2} P_S.$$

Then $P_L^\top A^{-1} P_S = X^\top C$ and $P_S^\top A^{-1} P_S = C^\top C$, so

$$T_2 \le \left\| X^\top C \left(\eta^{-1} I + C^\top C\right)^{-1} C^\top X \right\|_2 \le \|X\|_2^2 \cdot \left\| C \left(\eta^{-1} I + C^\top C\right)^{-1} C^\top \right\|_2 .$$

We claim $\|X\|_2 \le 1$. Indeed,

$$X^\top X = P_L^\top A^{-1} P_L = P_L^\top (P_L P_L^\top + I)^{-1} P_L = M(M+I)^{-1} \preceq I,$$

where $M = P_L^\top P_L \succeq 0$, and the last inequality holds since the eigenvalues of $M(M+I)^{-1}$ are $\lambda/(\lambda+1) \in [0,1)$. Therefore $\|X\|_2^2 \le 1$, and hence

$$T_2 \le \left\| C \left(\eta^{-1} I + C^\top C\right)^{-1} C^\top \right\|_2 .$$

Next, diagonalize $C^\top C$ and let $\sigma_{\max}^2 = \|C\|_2^2$ be its largest eigenvalue. The nonzero eigenvalues of $C(\eta^{-1} I + C^\top C)^{-1} C^\top$ are

$$\frac{\sigma_i^2}{\eta^{-1} + \sigma_i^2} = \frac{\eta \sigma_i^2}{1 + \eta \sigma_i^2},$$

so

$$\left\| C \left(\eta^{-1} I + C^\top C\right)^{-1} C^\top \right\|_2 = \frac{\eta \|C\|_2^2}{1 + \eta \|C\|_2^2} \le \eta \|C\|_2^2.$$

Finally, since $A \succeq I$, we have $\|C\|_2 = \|A^{-1/2} P_S\|_2 \le \|P_S\|_2$, and thus

$$T_2 \le \frac{\eta \|P_S\|_2^2}{1 + \eta \|P_S\|_2^2} \le \eta \|P_S\|_2^2.$$

It remains to control $\|P_S\|_2$. Since $P_S = \frac{1}{\sqrt{m}} W_S$ is Gaussian, standard spectral norm bounds imply that for any $\delta \in (0,1)$, with probability at least $1 - \delta$,

$$\|P_S\|_2 \le 1 + \sqrt{\frac{r-k}{m}} + \sqrt{\frac{\log(2/\delta)}{m}}.$$

In particular, if $r - k \le m$ and $m \ge \log(2/\delta)$, then on this event $\|P_S\|_2 \le 3$ and hence

$$T_2 \le 9\eta.$$

Choosing Parameters. Fix $\delta := \frac{1}{160}$ and assume $r - k \le m$ and $m \ge \log(2/\delta)$. This is possible by choosing r appropriately, e.g., $r = k + m$ or $r = 2k$ when $m \le k$. Then the above bound on T_2 holds with probability at least $1 - \delta$. By Theorem 1.4, the lower bound on T_1 holds with probability at least $3/80$. By the union bound, both events hold simultaneously with probability at least $3/80 - 1/160 = 1/32$. On this intersection event, using the bound from the T_1 part,

$$|\tilde{\tau}_\lambda(g,g) - \tau_\lambda(g,g)| \ge T_1 - T_2 \ge \min\left\{\frac{1}{16}\sqrt{\frac{k}{m}}, \frac{1}{4}\right\} - 9\eta.$$

We work in the nontrivial regime $\sqrt{k/m} \le 4$, so the minimum equals $\frac{1}{16}\sqrt{k/m}$. Now choose

$$\eta := \frac{\varepsilon}{288}.$$

If $m \le k/\varepsilon^2$, then $\sqrt{k/m} \ge \varepsilon$, and hence

$$\min\left\{\frac{1}{16}\sqrt{\frac{k}{m}}, \frac{1}{4}\right\} - 9\eta \ge \frac{1}{16}\varepsilon - \frac{9}{288}\varepsilon = \frac{1}{32}\varepsilon.$$

Therefore, with probability at least $1/32$,

$$|\tilde{\tau}_\lambda(g,g) - \tau_\lambda(g,g)| \ge \frac{1}{32}\varepsilon.$$

Recalling that $\tau_0(g,g) = 1$ and that $d_\lambda(F) = \Theta(k)$ for $\eta \ll 1$, this shows that whenever $m = o(d_\lambda(F)/\varepsilon^2)$, with constant probability there exists $g \in \mathrm{range}(F)$ such that

$$|\tilde{\tau}_\lambda(g,g) - \tau_\lambda(g,g)| = \Omega(\varepsilon)\tau_0(g,g),$$

as claimed. □

A.3 Proofs for Sec. 3.1.3 (Factorized Influence)

A.3.1 Proof of the Barrier of Unregularized Factorized Influence

We first record the factorized counterpart of the sharp barrier for exact preservation (Theorem 3.1) discussion in Sec. 3.1.3, i.e., Theorem 3.5. While Theorem 3.1 characterizes exact invariance for general sketches, the factorized sketch $P = P_A \otimes P_E$ admits a more explicit, factor-level injectivity condition. We restate Theorem 3.5 and prove it below:

Theorem. Let $A \succeq 0 \in \mathbb{R}^{d_A \times d_A}$, $E \succeq 0 \in \mathbb{R}^{d_E \times d_E}$, and $F := A \otimes E \succeq 0 \in \mathbb{R}^{(d_A d_E) \times (d_A d_E)}$. Let $r_A := \operatorname{rank}(A)$, $r_E := \operatorname{rank}(E)$, and $r := \operatorname{rank}(F) = r_A r_E$. Fix $P_A \in \mathbb{R}^{m_A \times d_A}$, $P_E \in \mathbb{R}^{m_E \times d_E}$, and define $P := P_A \otimes P_E \in \mathbb{R}^{(m_A m_E) \times (d_A d_E)}$. Then the following are equivalent:

(i) For all $g, g' \in \operatorname{range}(F)$, we have $\tilde{\tau}_0(g, g') = \tau_0(g, g')$.

(ii) P is injective on $\operatorname{range}(F)$, i.e., $\operatorname{rank}(PU) = r$ for *any* orthonormal basis $U \in \mathbb{R}^{(d_A d_E) \times r}$ of $\operatorname{range}(F)$.

(iii) P_A is injective on $\operatorname{range}(A)$ and P_E is injective on $\operatorname{range}(E)$. Equivalently, for orthonormal bases $U_A \in \mathbb{R}^{d_A \times r_A}$ of $\operatorname{range}(A)$ and $U_E \in \mathbb{R}^{d_E \times r_E}$ of $\operatorname{range}(E)$, we have $\operatorname{rank}(P_A U_A) = r_A$ and $\operatorname{rank}(P_E U_E) = r_E$.

In particular, $m_A \geq r_A$ and $m_E \geq r_E$ are necessary, hence $m = m_A m_E \geq r_A r_E = r$.

Proof. The equivalence between (i) and (ii) is exactly Theorem 3.1. It remains to relate (ii) and (iii) in the factorized setting. Let U_A and U_E be orthonormal bases of $\operatorname{range}(A)$ and $\operatorname{range}(E)$, respectively. Then $U := U_A \otimes U_E$ is an orthonormal basis of $\operatorname{range}(F)$. Using the mixed-product identity,

$$PU = (P_A \otimes P_E)(U_A \otimes U_E) = (P_A U_A) \otimes (P_E U_E).$$

Moreover, $\operatorname{rank}(X \otimes Y) = \operatorname{rank}(X)\operatorname{rank}(Y)$ for any matrices X, Y. Therefore,

$$\operatorname{rank}(PU) = \operatorname{rank}(P_A U_A)\operatorname{rank}(P_E U_E).$$

Since $\operatorname{rank}(P_A U_A) \le r_A$ and $\operatorname{rank}(P_E U_E) \le r_E$, we have $\operatorname{rank}(PU) = r_A r_E$ if and only if $\operatorname{rank}(P_A U_A) = r_A$ and $\operatorname{rank}(P_E U_E) = r_E$, which is equivalent to injectivity of P_A on $\operatorname{range}(A)$ and P_E on $\operatorname{range}(E)$.

The dimensional necessity $m_A \ge r_A$, $m_E \ge r_E$ follows immediately from $\operatorname{rank}(P_A U_A) \le \min\{m_A, r_A\}$ and $\operatorname{rank}(P_E U_E) \le \min\{m_E, r_E\}$. □

A.3.2 Proof of Factorized Resolvent Perturbation Concentration for Regularized Projection

This section proves the key technical lemma used in the factorized influence analysis in the main text (Theorem 3.6). The main technical challenges relative to the i.i.d. sketching setting are that, for a Kronecker sketch $P = P_A \otimes P_E$, the matrix $P^\top P$ decomposes into a sum of Kronecker-structured error terms rather than a single sample covariance, and P no longer satisfies the i.i.d. assumptions.

In the following, we prove the factorized version of Theorem 1.2:

Theorem 1.5 (Factorized covariance deviation for K-FAC)**.** *Let $F = A \otimes E \succeq 0$ and $P = P_A \otimes P_E$ be as above, and fix $\varepsilon, \delta \in (0,1)$. Assuming $\lambda \le \|A\|_2 \|E\|_2$, and define the rescaled regularization levels $\lambda_E := \lambda/\|E\|_2$ and $\lambda_A := \lambda/\|A\|_2$. If*

$$m_A = \Omega\left(\frac{d_{\lambda_E}(A) + \log(1/\delta)}{\varepsilon^2}\right), \qquad m_E = \Omega\left(\frac{d_{\lambda_A}(E) + \log(1/\delta)}{\varepsilon^2}\right),$$

then with probability at least $1 - 2\delta$,

$$\|B^\top (P^\top P - I) B\|_2 \le 2\varepsilon + 3\varepsilon^2.$$

Proof. Write

$$\Delta_A \coloneqq P_A^\top P_A - I_{d_A}, \qquad \Delta_E \coloneqq P_E^\top P_E - I_{d_E}.$$

Using $(X \otimes Y)^\top (X \otimes Y) = (X^\top X) \otimes (Y^\top Y)$, we have

$$P^\top P - I_{d_A d_E} = (P_A^\top P_A) \otimes (P_E^\top P_E) - I_{d_A} \otimes I_{d_E} = \Delta_A \otimes I_{d_E} + I_{d_A} \otimes \Delta_E + \Delta_A \otimes \Delta_E.$$

Therefore, by the triangle inequality,

$$\|B^\top (P^\top P - I) B\|_2 \le T_1 + T_2 + T_3, \tag{2}$$

where

$$T_1 \coloneqq \left\|B^\top (\Delta_A \otimes I_{d_E}) B\right\|_2, \quad T_2 \coloneqq \left\|B^\top (I_{d_A} \otimes \Delta_E) B\right\|_2, \quad T_3 \coloneqq \left\|B^\top (\Delta_A \otimes \Delta_E) B\right\|_2.$$

Bounding T_1. Let $A = U_A \Lambda_A U_A^\top$ and $E = U_E \Lambda_E U_E^\top$ be eigendecompositions with $\Lambda_A = \mathrm{diag}(\{\alpha_i\}_{i=1}^{d_A})$, $\Lambda_E = \mathrm{diag}(\{\gamma_j\}_{j=1}^{d_E})$, and U_A, U_E orthonormal. Then $F = A \otimes E$ is diagonalized by $U \coloneqq U_A \otimes U_E$, and

$$B = F^{1/2}(F + \lambda I)^{-1/2} = U D U^\top,$$

where D is diagonal with entries β_{ij} such that

$$\beta_{ij} \coloneqq \sqrt{\frac{\alpha_i \gamma_j}{\alpha_i \gamma_j + \lambda}}, \qquad (i,j) \in [d_A] \times [d_E].$$

Define $\widetilde{\Delta}_A \coloneqq U_A^\top \Delta_A U_A$. Then using the basic identity $(X \otimes Y)(Z \otimes W) = (XZ) \otimes (YW)$,

$$\begin{aligned} T_1 &= \|UDU^\top(\Delta_A \otimes I_{d_E})UDU^\top\|_2 \\ &= \|D(U_A^\top \otimes U_E^\top)(\Delta_A \otimes I_{d_E})(U_A \otimes U_E)D\|_2 \\ &= \|D(U_A^\top \Delta_A U_A \otimes I_{d_E})D\|_2 = \|D(\widetilde{\Delta}_A \otimes I_{d_E})D\|_2. \end{aligned}$$

The matrix $D(\widetilde{\Delta}_A \otimes I)D$ is not itself a Kronecker product, but it becomes block diagonal after a permutation of coordinates. Let $\Pi \in \{0,1\}^{(d_A d_E)\times(d_A d_E)}$ be the canonical commutation matrix satisfying

$$\Pi(X \otimes Y)\Pi^\top = Y \otimes X \qquad \text{for all conformable } X, Y.$$

Since Π is orthogonal, $\|M\|_2 = \|\Pi M \Pi^\top\|_2$ for any M. Thus,

$$T_1 = \|\Pi D(\widetilde{\Delta}_A \otimes I_{d_E})D\Pi^\top\|_2 = \|D_\Pi(I_{d_E} \otimes \widetilde{\Delta}_A)D_\Pi\|_2,$$

where $D_\Pi \coloneqq \Pi D \Pi^\top$ remains diagonal. The matrix $D_\Pi(I_{d_E} \otimes \widetilde{\Delta}_A)D_\Pi$ is block diagonal with d_E blocks; the j-th block (corresponding to the j-th eigenvalue γ_j) equals

$$D^{(j)}\widetilde{\Delta}_A D^{(j)}, \qquad D^{(j)} \coloneqq \operatorname{diag}(\{\beta_{ij}\}_{i=1}^{d_A}) = \operatorname{diag}\left(\left\{\sqrt{\frac{\alpha_i\gamma_j}{\alpha_i\gamma_j + \lambda}}\right\}_{i=1}^{d_A}\right).$$

Hence,

$$T_1 = \max_{j \in [d_E]} \|D^{(j)}\widetilde{\Delta}_A D^{(j)}\|_2. \tag{3}$$

We now compare each $D^{(j)}$ to a single dominating diagonal depending only on A. Since $\gamma_j \le \|E\|_2$ and $\alpha_i \ge 0$,[14]

$$\frac{\alpha_i\gamma_j}{\alpha_i\gamma_j + \lambda} \le \frac{\alpha_i}{\alpha_i + \lambda/\gamma_j} \le \frac{\alpha_i}{\alpha_i + \lambda/\|E\|_2} = \frac{\alpha_i}{\alpha_i + \lambda_E}.$$

[14]Note that the inequality holds trivially when $\gamma_j = 0$.

Define

$$D_A^{\max} := \operatorname{diag}\left(\left\{\sqrt{\frac{\alpha_i}{\alpha_i + \lambda_E}}\right\}_{i=1}^{d_A}\right).$$

Then for each j there exists a diagonal contraction $S^{(j)}$ such that

$$D^{(j)} = S^{(j)} D_A^{\max} = D_A^{\max} S^{(j)}, \qquad \|S^{(j)}\|_2 \le 1,$$

and therefore,

$$\|D^{(j)} \widetilde{\Delta}_A D^{(j)}\|_2 = \|S^{(j)} D_A^{\max} \widetilde{\Delta}_A D_A^{\max} S^{(j)}\|_2 \le \|D_A^{\max} \widetilde{\Delta}_A D_A^{\max}\|_2.$$

Combining with Eq. (3) yields

$$T_1 \le \|D_A^{\max} \widetilde{\Delta}_A D_A^{\max}\|_2.$$

Finally, note that

$$D_A^{\max} \widetilde{\Delta}_A D_A^{\max} = (U_A D_A^{\max})^\top (P_A^\top P_A - I_{d_A})(U_A D_A^{\max}).$$

Let $M_A := U_A D_A^{\max}$. Applying Theorem 1.2 to M_A and sketching P_A (with failure probability δ) gives that when

$$m_A = \Omega\left(\frac{r(M_A^\top M_A) + \log(1/\delta)}{\varepsilon^2}\right),$$

we have $T_1 \le \varepsilon$ with probability at least $1 - \delta$. It remains to identify $r(M_A^\top M_A)$. Since $M_A^\top M_A = (D_A^{\max})^2$ is diagonal with spectral norm at most 1,

$$r(M_A^\top M_A) = \frac{\operatorname{tr}((D_A^{\max})^2)}{\|(D_A^{\max})^2\|_2} = \operatorname{tr}((D_A^{\max})^2) = \sum_{i=1}^{d_A} \frac{\alpha_i}{\alpha_i + \lambda_E} = d_{\lambda_E}(A).$$

Thus, under the stated condition on m_A, with probability at least $1-\delta$,

$$T_1 \le \varepsilon. \tag{4}$$

Bounding T_2. The bound for T_2 is identical by symmetry (and is in fact simpler because $I_{d_A} \otimes \widetilde{\Delta}_E$ is already block diagonal in the A-first ordering). Specifically, define $\widetilde{\Delta}_E \coloneqq U_E^\top \Delta_E U_E$ and

$$D_E^{\max} \coloneqq \operatorname{diag}\left(\left\{ \sqrt{\frac{\gamma_j}{\gamma_j + \lambda_A}} \right\}_{j=1}^{d_E} \right), \qquad M_E \coloneqq U_E D_E^{\max}.$$

Applying Theorem 1.2 to M_E and P_E yields that, when

$$m_E = \Omega\left(\frac{d_{\lambda_A}(E) + \log(1/\delta)}{\varepsilon^2} \right),$$

we have with probability at least $1-\delta$,

$$T_2 \le \varepsilon. \tag{5}$$

Bounding T_3. We show that the diagonal D is dominated by a Kronecker product of the dominating diagonals $D_A^{\max}$ and $D_E^{\max}$, up to a universal constant, provided $\lambda \le \|A\|_2 \|E\|_2$. For each (i,j),

$$\beta_{ij}^2 = \frac{\alpha_i \gamma_j}{\alpha_i \gamma_j + \lambda}.$$

We claim that

$$\frac{\alpha_i \gamma_j}{\alpha_i \gamma_j + \lambda} \le 3 \cdot \frac{\alpha_i}{\alpha_i + \lambda_E} \cdot \frac{\gamma_j}{\gamma_j + \lambda_A}. \tag{6}$$

Indeed, Eq. (6) is equivalent (after taking reciprocals of positive quantities) to

$$(\alpha_i + \lambda_E)(\gamma_j + \lambda_A) \le 3(\alpha_i \gamma_j + \lambda).$$

Expanding the left-hand side gives

$$(\alpha_i + \lambda_E)(\gamma_j + \lambda_A) = \alpha_i\gamma_j + \alpha_i\lambda_A + \gamma_j\lambda_E + \lambda_A\lambda_E.$$

Using $\alpha_i \le \|A\|_2$, $\gamma_j \le \|E\|_2$, and the definitions $\lambda_A = \lambda/\|A\|_2$, $\lambda_E = \lambda/\|E\|_2$, we obtain

$$\alpha_i\lambda_A \le \lambda, \qquad \gamma_j\lambda_E \le \lambda, \qquad \lambda_A\lambda_E = \frac{\lambda^2}{\|A\|_2\|E\|_2} \le \lambda,$$

where the last inequality uses the assumption $\lambda \le \|A\|_2\|E\|_2$. Therefore,

$$(\alpha_i + \lambda_E)(\gamma_j + \lambda_A) \le \alpha_i\gamma_j + 3\lambda \le 3(\alpha_i\gamma_j + \lambda),$$

which proves Eq. (6). Taking square-roots yields

$$\beta_{ij} \le \sqrt{3}\sqrt{\frac{\alpha_i}{\alpha_i + \lambda_E}}\sqrt{\frac{\gamma_j}{\gamma_j + \lambda_A}}.$$

Therefore, there exists a diagonal contraction S such that

$$D = \sqrt{3}S(D_A^{\max} \otimes D_E^{\max}) = \sqrt{3}(D_A^{\max} \otimes D_E^{\max})S, \qquad \|S\|_2 \le 1,$$

and therefore

$$\begin{aligned} T_3 &= 3\|S(D_A^{\max} \otimes D_E^{\max})(\widetilde{\Delta}_A \otimes \widetilde{\Delta}_E)(D_A^{\max} \otimes D_E^{\max})S\|_2 \\ &\le 3\|(D_A^{\max}\widetilde{\Delta}_A D_A^{\max}) \otimes (D_E^{\max}\widetilde{\Delta}_E D_E^{\max})\|_2. \end{aligned}$$

Since $\|X \otimes Y\|_2 = \|X\|_2\|Y\|_2$, this becomes

$$T_3 \le 3\|D_A^{\max}\widetilde{\Delta}_A D_A^{\max}\|_2\|D_E^{\max}\widetilde{\Delta}_E D_E^{\max}\|_2.$$

On the event where both Eq. (4) and Eq. (5) hold, we obtain

$$T_3 \leq 3\varepsilon^2. \tag{7}$$

Putting together. By Eqs. (2), (4), (5) and (7), on the intersection of the two concentration events (one for P_A, one for P_E),

$$\|B^\top (P^\top P - I)B\|_2 \leq \varepsilon + \varepsilon + 3\varepsilon^2 = 2\varepsilon + 3\varepsilon^2.$$

The two concentration events each fail with probability at most δ, so by a union bound, the intersection holds with probability at least $1 - 2\delta$. This completes the proof. □

A.3.3 Note on Proof of Theorem 3.6

We note that while Theorem 1.5 is stated with failure probability 2δ and deviation level $2\varepsilon + 3\varepsilon^2$, in the proof of Theorem 3.6, we require it to be with failure probability δ and deviation level ε. This is only for notational convenience: given target parameters (ε, δ), one may apply the theorem with $\epsilon := \varepsilon/10$ and $\eta := \delta/2$, which yields probability at least $1 - 2\eta = 1 - \delta$ and deviation at most $2\epsilon + 3\epsilon^2 \leq \varepsilon/2 \leq \varepsilon$ for $\varepsilon \in (0, 1)$.

A.4 Proofs for Sec. 3.2.1 (Leakage of Projection)

In this section, we prove Theorem 3.8, which we first repeat the statement for convenience:

Theorem. Let $\{g'_j\}_{j=1}^k \subset \mathbb{R}^d$, and for each j let $g'_{j,\perp}$ denote the orthogonal projection of g'_j onto $\ker(F)$. Let $k' = \dim(\mathrm{span}(\{g'_{j,\perp}\}_{j=1}^k))$. For any $\varepsilon, \delta \in (0, 1)$, if

$$m = \Omega\left(\frac{r + \min\left(\log(k/\delta), k' + \log(1/\delta)\right)}{\varepsilon^2}\right),$$

then with probability at least $1 - \delta$, the following holds for all $j \in \{1, \ldots, k\}$:

- **Unregularized**: For $T_j := (Pg)^\top (PFP^\top)^\dagger (Pg'_{j,\perp})$, we have

$$|T_j| \leq \varepsilon \frac{\|g\|_2 \|g'_{j,\perp}\|_2}{\lambda^+_{\min}(F)},$$

  where $\lambda^+_{\min}(F)$ denotes the smallest non-zero eigenvalue of F.

- **Regularized**: For $T_{\lambda,j} := (Pg)^\top (PFP^\top + \lambda I)^{-1} (Pg'_{j,\perp})$, we have

$$|T_{\lambda,j}| \leq \varepsilon \|g\|_2 \|g'_{j,\perp}\|_2 \left(\frac{1}{\lambda} + \frac{2\|F\|_2}{\lambda^2} \right)$$

A.4.1 Proof Plan for Theorem 3.8

The main organizing step is a deterministic reduction: Theorem 1.6 shows that *both* the regularized and unregularized leakage bounds follow once the sketch P satisfies two concentration conditions with respect to an orthonormal basis U of range(F): (i) subspace stability on range(F), $\|U^\top (P^\top P - I_d) U\|_2 \leq \varepsilon$, and (ii) cross-term control between range(F) and the kernel direction(s), $\|U^\top (P^\top P - I_d) g'_{j,\perp}\|_2 \leq \varepsilon \|g'_{j,\perp}\|_2$ for each j. Indeed, this is shown formally in Theorem 1.6.

Lemma 1.6. *Let $\{g'_j\}_{j=1}^k \subset \mathbb{R}^d$, and for each j let $g'_{j,\perp}$ denote the orthogonal projection of g'_j onto* ker(F)*. Fix a realization of P, and let $U \in \mathbb{R}^{d \times r}$ be an orthonormal basis for* range(F)*. Assume that for some $\varepsilon \in (0,1)$, the following two inequalities hold:*

(i) $\|U^\top (P^\top P - I_d) U\|_2 \leq \varepsilon$,

(ii) $\|U^\top (P^\top P - I_d) g'_{j,\perp}\|_2 \leq \varepsilon \|g'_{j,\perp}\|_2$ *for every* $j \in \{1, \dots, k\}$.

Then for any fixed $g \in$ range(F)*, the following bounds hold* simultaneously for all $j \in \{1, \dots, k\}$*:*

$$\left| (Pg)^\top (PFP^\top)^\dagger (Pg'_{j,\perp}) \right| \leq \varepsilon \frac{1+\varepsilon}{(1-\varepsilon)^2} \cdot \frac{\|g\|_2 \|g'_{j,\perp}\|_2}{\lambda^+_{\min}(F)}.$$

and

$$\left|(Pg)^\top (PFP^\top + \lambda I)^{-1}(Pg'_{j,\perp})\right| \le \varepsilon \|g\|_2 \|g'_{j,\perp}\|_2 \left(\frac{1}{\lambda} + \frac{2\|F\|_2}{\lambda^2}\right).$$

Proof. We prove the unregulairzed case first.

Unregularized Case. Fix j. Using $\mathrm{range}(PFP^\top) = \mathrm{range}(PU)$, let

$$\Pi_{PU} \coloneqq PU(U^\top P^\top PU)^{-1}(PU)^\top$$

denote the orthogonal projector onto $\mathrm{range}(PU)$. Then

$$(Pg)^\top (PFP^\top)^\dagger (Pg'_{j,\perp}) = g^\top P^\top (PFP^\top)^\dagger \Pi_{PU} P g'_{j,\perp},$$

and hence

$$\left|(Pg)^\top (PFP^\top)^\dagger (Pg'_{j,\perp})\right| \le \|(PFP^\top)^\dagger\|_2 \|Pg\|_2 \|\Pi_{PU} P g'_{j,\perp}\|_2.$$

We bound the three terms on the right-hand side.

First, we bound $\|(PFP^\top)^\dagger\|_2$. Write the compact eigendecomposition $F = U\Sigma U^\top$, where $\Sigma \succ 0$ is diagonal and $\|\Sigma^{-1}\|_2 = 1/\lambda^+_{\min}(F)$. Since $PFP^\top = (PU)\Sigma(PU)^\top$, we have

$$\|(PFP^\top)^\dagger\|_2 = \|(PU)^\dagger\|_2^2 \|\Sigma^{-1}\|_2 = \frac{1}{\sigma_{\min}(PU)^2} \cdot \frac{1}{\lambda^+_{\min}(F)}.$$

Moreover, assumption (i) implies that all eigenvalues of $U^\top P^\top PU = (PU)^\top (PU)$ lie in $[1-\varepsilon, 1+\varepsilon]$, hence $\sigma_{\min}(PU)^2 \ge 1-\varepsilon$ and

$$\|(PFP^\top)^\dagger\|_2 \le \frac{1}{(1-\varepsilon)\lambda^+_{\min}(F)}.$$

To bound $\|Pg\|_2$, write $g = Uh$, we have $\|Pg\|_2 \le \sqrt{1+\varepsilon}\|g\|_2$ since

$$\|Pg\|_2^2 = h^\top (U^\top P^\top PU) h \le (1+\varepsilon)\|h\|_2^2 = (1+\varepsilon)\|g\|_2^2.$$

Finally, to bound $\|\Pi_{PU} P g'_{j,\perp}\|_2$, with $U^\top g'_{j,\perp} = 0$, we have $U^\top P^\top P g'_{j,\perp} = U^\top (P^\top P - I_d) g'_{j,\perp}$, hence

$$\begin{aligned}\|\Pi_{PU} P g'_{j,\perp}\|_2 &= \left\| PU(U^\top P^\top PU)^{-1} U^\top P^\top P g'_{j,\perp} \right\|_2 \\ &\le \|PU\|_2 \|(U^\top P^\top PU)^{-1}\|_2 \|U^\top (P^\top P - I_d) g'_{j,\perp}\|_2.\end{aligned}$$

By assumption (i), $\|PU\|_2 = \sigma_{\max}(PU) \le \sqrt{1+\varepsilon}$ and $\|(U^\top P^\top PU)^{-1}\|_2 \le 1/(1-\varepsilon)$. By assumption (ii), $\|U^\top (P^\top P - I_d) g'_{j,\perp}\|_2 \le \varepsilon \|g'_{j,\perp}\|_2$. Therefore,

$$\|\Pi_{PU} P g'_{j,\perp}\|_2 \le \varepsilon \frac{\sqrt{1+\varepsilon}}{1-\varepsilon} \|g'_{j,\perp}\|_2.$$

Combining the bounds yields

$$|(Pg)^\top (PFP^\top)^\dagger (P g'_{j,\perp})| \le \varepsilon \frac{1+\varepsilon}{(1-\varepsilon)^2} \cdot \frac{\|g\|_2 \|g'_{j,\perp}\|_2}{\lambda^+_{\min}(F)}.$$

Regularized Case. Fix $g \in \mathrm{range}(F)$ and j. Write $g = Uh$. Set $V := PF^{1/2}$, so that $PFP^\top = VV^\top$. By the Woodbury identity,

$$(VV^\top + \lambda I)^{-1} = \frac{1}{\lambda} I - \frac{1}{\lambda^2} V \Big(I + \frac{1}{\lambda} V^\top V \Big)^{-1} V^\top.$$

Therefore, writing $T_{\lambda,j} = (Pg)^\top (VV^\top + \lambda I)^{-1} (P g'_{j,\perp})$, we have the decomposition $T_{\lambda,j} = T^{(1)}_{\lambda,j} - T^{(2)}_{\lambda,j}$ where

$$T^{(1)}_{\lambda,j} = \frac{1}{\lambda} g^\top P^\top P g'_{j,\perp}, \qquad T^{(2)}_{\lambda,j} = \frac{1}{\lambda^2} g^\top P^\top P F^{1/2} \Big(I + \frac{1}{\lambda} V^\top V \Big)^{-1} F^{1/2} P^\top P g'_{j,\perp}.$$

Since $g^\top g'_{j,\perp} = 0$,

$$|T^{(1)}_{\lambda,j}| = \frac{1}{\lambda} |g^\top (P^\top P - I_d) g'_{j,\perp}| = \frac{1}{\lambda} |h^\top U^\top (P^\top P - I_d) g'_{j,\perp}| \le \frac{\varepsilon}{\lambda} \|g\|_2 \|g'_{j,\perp}\|_2,$$

using Cauchy–Schwarz and assumption (ii).

Next, we bound $T^{(2)}_{\lambda,j}$. Note that $V^\top V = F^{1/2}P^\top PF^{1/2} \succeq 0$, so $I + \frac{1}{\lambda}V^\top V \succeq I$, which implies $\left\|(I + \frac{1}{\lambda}V^\top V)^{-1}\right\|_2 \leq 1$. Moreover, since $\text{range}(F^{1/2}) = \text{range}(F)$, we have $F^{1/2} = \Pi_F F^{1/2} = F^{1/2}\Pi_F$, and hence we may insert Π_F on both sides of each $F^{1/2}$ factor. Using sub-multiplicativity and $\|F^{1/2}\|_2^2 = \|F\|_2$, we obtain

$$|T^{(2)}_{\lambda,j}| \leq \frac{1}{\lambda^2}\|\Pi_F P^\top Pg\|_2 \|F\|_2 \|\Pi_F P^\top P g'_{j,\perp}\|_2.$$

Since $\Pi_F = UU^\top$ and $g = Uh$,

$$\|\Pi_F P^\top Pg\|_2 = \|U(U^\top P^\top PU)h\|_2 \leq \|U^\top P^\top PU\|_2 \|g\|_2 \leq (1+\varepsilon)\|g\|_2,$$

where we used $U^\top P^\top PU = I_r + U^\top(P^\top P - I_d)U$ and assumption (i). Moreover, since $U^\top g'_{j,\perp} = 0$,

$$\|\Pi_F P^\top P g'_{j,\perp}\|_2 = \|U^\top P^\top P g'_{j,\perp}\|_2 = \|U^\top(P^\top P - I_d)g'_{j,\perp}\|_2 \leq \varepsilon \|g'_{j,\perp}\|_2,$$

by assumption (ii). Hence

$$|T^{(2)}_{\lambda,j}| \leq \frac{\|F\|_2}{\lambda^2}(1+\varepsilon)\varepsilon\|g\|_2\|g'_{j,\perp}\|_2.$$

Combining the two pieces and using $\varepsilon \leq 1$ gives

$$|T_{\lambda,j}| \leq \varepsilon\|g\|_2\|g'_{j,\perp}\|_2 \left(\frac{1}{\lambda} + \frac{2\|F\|_2}{\lambda^2}\right).$$

□

Thus, the remaining work in the proof is to verify these two conditions in the single-gradient and multi-gradient regimes. We break the proof into the following cases:

1. For a single kernel component $g'_\perp \in \ker(F)$:

- Theorem 1.7: prove the bound for unregularized and regularized case.

2. Extend both cases to $\{g'_{j,\perp}\}_{j=1}^{k} \subseteq \ker(F)$ with $k' = \dim(\mathrm{span}(\{g'_{j,\perp}\}_{j=1}^{k}))$:

   - Theorem 1.8: subspace argument with $m = \Omega\left(\frac{r+k'+\log(1/\delta)}{\varepsilon^2}\right)$.
   - Theorem 1.9: union-bound argument with $m = \Omega\left(\frac{r+\log(k/\delta)}{\varepsilon^2}\right)$.

We now start the proof.

A.4.2 Proof of Single Test Gradient Leakage

Proposition 1.7. *Assume $g \in \mathrm{range}(F)$ and let $g' \in \mathbb{R}^d$. For any $\varepsilon, \delta \in (0,1)$, if $m = \Omega(\varepsilon^{-2}(r + \log(1/\delta)))$, then with probability at least $1-\delta$,*

1. ***Unregularized**: Let $T := (Pg)^\top (PFP^\top)^\dagger (Pg'_\perp)$, then*

$$|T| \le \varepsilon \frac{\|g\|_2 \|g'_\perp\|_2}{\lambda^+_{\min}(F)},$$

   where $\lambda^+_{\min}(F)$ denotes the smallest non-zero eigenvalue of F.

2. ***Regularized**: Let $T_\lambda := (Pg)^\top (PFP^\top + \lambda I)^{-1} (Pg'_\perp)$, then*

$$|T_\lambda| \le \varepsilon \|g\|_2 \|g'_\perp\|_2 \left(\frac{1}{\lambda} + \frac{2\|F\|_2}{\lambda^2}\right).$$

Proof. From Theorem 1.6, it suffices to verify conditions (i) and (ii).

Let $S := \mathrm{span}(\mathrm{range}(F) \cup \{g'_\perp\})$, so $\dim(S) = r+1$, and let $W \in \mathbb{R}^{d \times (r+1)}$ be an orthonormal basis for S. Fix any $\eta \in (0,1)$ and define the event

$$\mathcal{E}(\eta) := \left\{ \left\| W^\top (P^\top P - I_d) W \right\|_2 \le \eta \right\}.$$

On $\mathcal{E}(\eta)$, for any orthonormal basis $U \in \mathbb{R}^{d \times r}$ of $\mathrm{range}(F) \subseteq S$ there exists $R \in \mathbb{R}^{(r+1) \times r}$

with $R^\top R = I_r$ such that $U = WR$. Thus,

$$\|U^\top (P^\top P - I_d) U\|_2 = \|R^\top W^\top (P^\top P - I_d) W R\|_2 \le \eta.$$

Moreover, since $g'_\perp \in S$, we have $g'_\perp = WW^\top g'_\perp$ and $\|W^\top g'_\perp\|_2 = \|g'_\perp\|_2$, and hence

$$\|U^\top (P^\top P - I_d) g'_\perp\|_2 = \|R^\top W^\top (P^\top P - I_d) W W^\top g'_\perp\|_2 \le \eta \|g'_\perp\|_2.$$

Therefore, on $\mathcal{E}(\eta)$ the assumptions of Theorem 1.6 hold with parameter η.

Unregularized. By Theorem 1.2 applied to S, if $m = \Omega(((r+1) + \log(1/\delta))/\eta^2)$, then $\mathbb{P}(\mathcal{E}(\eta)) \ge 1 - \delta$. Taking $\eta = \varepsilon/4$ and applying Theorem 1.6 yields

$$|T| \le \frac{\varepsilon}{4} \cdot \frac{1 + \varepsilon/4}{(1 - \varepsilon/4)^2} \cdot \frac{\|g\|_2 \|g'_\perp\|_2}{\lambda^+_{\min}(F)}.$$

As in the previous argument, $\frac{1+\varepsilon/4}{(1-\varepsilon/4)^2} \le \frac{20}{9}$, hence the prefactor is $\le \varepsilon$.

Regularized. Taking $\eta = \varepsilon$ and applying Theorem 1.6 gives

$$|T_\lambda| \le \varepsilon \|g\|_2 \|g'_\perp\|_2 \left(\frac{1}{\lambda} + \frac{2\|F\|_2}{\lambda^2} \right).$$

□

This result shows that, in the unregularized case, the kernel leakage term decays at rate $O(m^{-1/2})$, with constants that depend on the smallest non-zero eigenvalue of F. On the other hand, in the regularized case, the kernel leakage term also decays at rate $O(m^{-1/2})$, but with constants depending on $\|F\|_2$ and the regularization parameter λ. This dependence reflects the sensitivity of the pseudoinverse to near-degeneracies in the spectrum of F.

A.4.3 Proof of Multiple Test Gradients Leakage

Having established the deterministic reduction in Theorem 1.6, we now show how to enforce its two assumptions uniformly over multiple test gradients. First, we observe that the previous analysis naturally generalizes by considering the subspace spanned by all test gradients.

Proposition 1.8. *Let $\{g'_j\}_{j=1}^k \subset \mathbb{R}^d$, and for each j let $g'_{j,\perp}$ denote the orthogonal projection of g'_j onto $\ker(F)$. Let $k' = \dim(\mathrm{span}(\{g'_{j,\perp}\}_{j=1}^k))$. For any $\varepsilon, \delta \in (0,1)$, if*

$$m = \Omega\left(\frac{r + k' + \log(1/\delta)}{\varepsilon^2}\right),$$

then with probability at least $1-\delta$, the leakage bounds in Theorem 1.7 hold simultaneously for all $j \in \{1, \dots, k\}$.

Proof. Let $S := \mathrm{span}(\mathrm{range}(F) \cup \{g'_{j,\perp}\}_{j=1}^k)$, so that $\dim(S) = r + k'$, and let $W \in \mathbb{R}^{d \times (r+k')}$ be an orthonormal basis for S. By Theorem 1.2, with probability at least $1-\delta$,

$$\left\| W^\top (P^\top P - I_d) W \right\|_2 \le \varepsilon,$$

provided that $m = \Omega(\varepsilon^{-2}(r + k' + \log(1/\delta)))$. On this event, for all $x, y \in S$,

$$\left| x^\top (P^\top P - I_d) y \right| = \left| (W^\top x)^\top W^\top (P^\top P - I_d) W (W^\top y) \right| \le \varepsilon \|x\|_2 \|y\|_2.$$

Now let $U \in \mathbb{R}^{d \times r}$ be an orthonormal basis for $\mathrm{range}(F)$. Since $\mathrm{range}(F) \subseteq S$, the columns of U are contained in S, and hence

$$\left\| U^\top (P^\top P - I_d) U \right\|_2 \le \left\| W^\top (P^\top P - I_d) W \right\|_2 \le \varepsilon.$$

Moreover, for each j, using that U has orthonormal columns and $\mathrm{range}(F) \subseteq S$, we have

$$\begin{aligned}\left\|U^\top (P^\top P - I_d) g'_{j,\perp}\right\|_2 &= \sup_{\substack{a \in \mathbb{R}^r \\ \|a\|_2 = 1}} \left|a^\top U^\top (P^\top P - I_d) g'_{j,\perp}\right| \\ &= \sup_{\substack{x \in \mathrm{range}(F) \\ \|x\|_2 = 1}} \left|x^\top (P^\top P - I_d) g'_{j,\perp}\right| \le \varepsilon \|g'_{j,\perp}\|_2,\end{aligned}$$

where the last inequality applies the bilinear bound above with $x \in \mathrm{range}(F) \subseteq S$ and $y = g'_{j,\perp} \in S$. Thus, the assumptions of Theorem 1.6 hold simultaneously for all j, and the corollary follows by applying Theorem 1.6. □

While Theorem 1.8 is effective when the test gradients are low-dimensional, as $g'_j \in \mathbb{R}^d$ lies in high dimension, it is almost certain that k' will be large, and most likely $k' \approx k$. In this case, by directly controlling the concentration of the bilinear form, we can obtain a bound that scales only logarithmically with the *number* of test gradients.

Proposition 1.9. *Let $\{g'_j\}_{j=1}^k \subset \mathbb{R}^d$, and for each j let $g'_{j,\perp}$ denote the orthogonal projection of g'_j onto $\ker(F)$. For any $\varepsilon, \delta \in (0,1)$, if*

$$m = \Omega\left(\frac{r + \log(k/\delta)}{\varepsilon^2}\right),$$

then with probability at least $1 - \delta$, the leakage bounds in Theorem 1.7 hold for all $j \in \{1, \dots, k\}$.

Proof. Let $U \in \mathbb{R}^{d \times r}$ be an orthonormal basis for $\mathrm{range}(F)$. We will verify the two assumptions of Theorem 1.6 uniformly over $\{g'_{j,\perp}\}_{j=1}^k$. Since Theorem 1.6 incurs a benign factor $\frac{1+\varepsilon}{(1-\varepsilon)^2}$ in the unregularized case, we will run the concentration argument below with accuracy parameter $\varepsilon/4$; the resulting constant-factor strengthening is absorbed by the $\Omega(\cdot)$ sample complexity.

Controlling $\|U^\top(P^\top P - I_d)U\|_2$. By Theorem 1.2 applied to the r-dimensional subspace range(F), with probability at least $1 - \delta/2$,

$$\|U^\top(P^\top P - I_d)U\|_2 \le \varepsilon,$$

provided that $m = \Omega(\varepsilon^{-2}(r + \log(2/\delta)))$.

Controlling $\|U^\top(P^\top P - I_d)g'_{j,\perp}\|_2$ for all j. Fix $g'_\perp \in \ker(F)$ with $\|g'_\perp\|_2 = 1$. Note that

$$\|U^\top(P^\top P - I_d)g'_\perp\|_2 = \sup_{a \in S^{r-1}} |(Ua)^\top(P^\top P - I_d)g'_\perp| = \sup_{x \in US^{r-1}} |x^\top(P^\top P - I_d)g'_\perp|,$$

where $US^{r-1} = \{Ua : a \in \mathbb{R}^r, \|a\|_2 = 1\}$ is the unit sphere in range(F).

By the polarization identity $x^\top y = \frac{1}{4}(\|x+y\|_2^2 - \|x-y\|_2^2)$, the bilinear form can be written as:

$$x^\top(P^\top P - I_d)g'_\perp = \frac{1}{4}\left(\|P(x+g'_\perp)\|_2^2 - \|x+g'_\perp\|_2^2\right) - \frac{1}{4}\left(\|P(x-g'_\perp)\|_2^2 - \|x-g'_\perp\|_2^2\right).$$

To bound this uniformly over $x \in US^{r-1}$, define the set $T = T_+ \cup T_-$, where $T_\pm = \{x \pm g'_\perp : x \in US^{r-1}\}$. It follows that

$$\sup_{x \in US^{r-1}} |x^\top(P^\top P - I_d)g'_\perp| \le \frac{1}{2}\sup_{z \in T} |\|Pz\|_2^2 - \|z\|_2^2|.$$

Define the sub-Gaussian stochastic process $Y_z = \|Pz\|_2 - \|z\|_2$ for $z \in T$, similar to the proof of Theorem 1.1. Applying the Talagrand comparison inequality [225, Theorem 3.2], with probability at least $1 - 2e^{-u}$,

$$\sup_{z \in T} |\|Pz\|_2 - \|z\|_2| \le \frac{C}{\sqrt{m}}(\gamma(T) + \sqrt{u} \cdot \mathrm{rad}(T)).$$

We analyze the radius and Gaussian complexity of T:

- $\mathrm{rad}(T)$: For any $z \in T$, $z = x \pm g'_\perp$. Since $x \perp g'_\perp$ as $x \in \mathrm{range}(F)$ and $g'_\perp \in \ker(F)$, the Pythagorean theorem gives $\|z\|_2^2 = \|x\|_2^2 + \|g'_\perp\|_2^2 = 1 + 1 = 2$, giving $\mathrm{rad}(T) = \sqrt{2} = O(1)$.

- $\gamma(T)$: By definition, $\gamma(T) = \mathbb{E}[\sup_{z \in T} |\langle h, z \rangle|]$ for $h \sim \mathcal{N}(0, I_d)$. For $z = x \pm g'_\perp$, we have $\langle h, x \pm g'_\perp \rangle = \langle h, x \rangle \pm \langle h, g'_\perp \rangle$. Thus,

$$\gamma(T) \leq \mathbb{E}\left[\sup_{x \in US^{r-1}} |\langle h, x \rangle|\right] + \mathbb{E}[|\langle h, g'_\perp \rangle|].$$

  The first term is the Gaussian complexity of the unit sphere in an r-dimensional subspace, which is bounded by $\sqrt{r}$. The second term is $\mathbb{E}[Z]$ for $Z \sim \mathcal{N}(0, 1)$, which is $\sqrt{2/\pi}$. Overall, $\gamma(T) \leq \sqrt{r} + \sqrt{2/\pi} \lesssim \sqrt{r}$.

With again $|a^2 - b^2| \leq |a - b|(|a - b| + 2b)$ with $a = \|Pz\|_2$ and $b = \|z\|_2 = \sqrt{2}$, we have

$$\sup_{z \in T} |\|Pz\|_2^2 - \|z\|_2^2| \leq \frac{C}{\sqrt{m}}(\gamma(T) + \sqrt{u}\,\mathrm{rad}(T)) \left(\frac{C}{\sqrt{m}}(\gamma(T) + \sqrt{u}\,\mathrm{rad}(T)) + 2\,\mathrm{rad}(T)\right).$$

Distributing the terms and substituting $\mathrm{rad}(T) = \sqrt{2}$ and $\gamma(T) \leq \sqrt{r} + 1$, we have

$$\sup_{z \in T} |\|Pz\|_2^2 - \|z\|_2^2| \leq C\left(\frac{r + u}{m} + \sqrt{\frac{r + u}{m}}\right).$$

Setting $u = \log(4k/\delta)$ ensures that $2e^{-u} = \delta/(2k)$. Hence, for a fixed $g'_\perp$, we have $\|U^\top(P^\top P - I_d)g'_\perp\|_2 \leq \varepsilon$ with probability at least $1 - \delta/(2k)$, provided that $m = \Omega((r + \log(k/\delta))/\varepsilon^2)$. By a union bound over $j \in \{1, \ldots, k\}$, the bound holds simultaneously for all k test gradients with probability at least $1 - \delta/2$.

Finally, taking a union bound over the two failure events (the subspace event and the k bilinear events), the same argument (with ε replaced by $\varepsilon/4$) yields that with probability at least $1 - \delta$, $\left\|U^\top(P^\top P - I_d)U\right\|_2 \leq \varepsilon/4$ and

$$\left\|U^\top(P^\top P - I_d)g'_{j,\perp}\right\|_2 \leq (\varepsilon/4)\|g'_{j,\perp}\|_2 \quad \text{for all } j \in \{1, \ldots, k\}.$$

On this event, we apply Theorem 1.6 with parameter $\varepsilon/4$. The regularized leakage bound then holds with prefactor $\varepsilon/4 \le \varepsilon$. For the unregularized leakage bound, we obtain

$$\left|(Pg)^\top (PFP^\top)^\dagger (Pg'_{j,\perp})\right| \le \frac{\varepsilon}{4} \cdot \frac{1+\varepsilon/4}{(1-\varepsilon/4)^2} \cdot \frac{\|g\|_2 \|g'_{j,\perp}\|_2}{\lambda^+_{\min}(F)} \le \varepsilon \cdot \frac{\|g\|_2 \|g'_{j,\perp}\|_2}{\lambda^+_{\min}(F)},$$

using $\frac{1+\varepsilon/4}{(1-\varepsilon/4)^2} \le \frac{20}{9}$ as in Theorem 1.7. This completes the proof. □

A.5 Proofs for Sec. 3.2.2 (Leakage of Factorized Influence)

In this subsection, we extend the leakage analysis in Appendix A.4 to the factorized influence setting. We consider curvature matrices of the form

$$F = A \otimes E \in \mathbb{R}^{(d_A d_E)\times(d_A d_E)},$$

where $A \succeq 0$ and $E \succeq 0$. We analyze a factorized sketch

$$P = P_A \otimes P_E, \qquad P_A \in \mathbb{R}^{m_A \times d_A}, P_E \in \mathbb{R}^{m_E \times d_E},$$

so that $P \in \mathbb{R}^{(m_A m_E)\times(d_A d_E)} = \mathbb{R}^{m\times d}$ with $m = m_A m_E$ and $d = d_A d_E$. Throughout, we assume P_A and P_E are both oblivious sketches as defined in Theorem 3.2. We will show that the only new work needed is to bound the cross-term quantity $\|U^\top (P^\top P - I) g'_\perp\|_2$ (for kernel components $g'_\perp \in \ker(F)$) appearing in Theorem 1.6 via factor-level primitive bounds.

Theorem. Let $A, E \succeq 0$ and $F := A \otimes E$, and let $P = P_A \otimes P_E$ with $P_A \in \mathbb{R}^{m_A \times d_A}$ and $P_E \in \mathbb{R}^{m_E \times d_E}$. Let $r_A := \operatorname{rank}(A)$, $r_E := \operatorname{rank}(E)$, and $r := \operatorname{rank}(F) = r_A r_E$.

Let $\{g'_j\}_{j=1}^k \subset \mathbb{R}^{d_A d_E}$ be test gradients of the form $g'_j = a'_j \otimes e'_j$. For each j, define the kernel component $g'_{j,\perp} := \Pi_{\ker(F)} g'_j$. Write $a'_j = a'_{j,\|} + a'_{j,\perp}$ with $a'_{j,\|} \in \operatorname{range}(A)$ and $a'_{j,\perp} \perp \operatorname{range}(A)$, and similarly $e'_j = e'_{j,\|} + e'_{j,\perp}$. Define $k_A := \sum_{j=1}^k \mathbb{1}(a'_{j,\perp} \neq 0)$, $k_E := \sum_{j=1}^k \mathbb{1}(e'_{j,\perp} \neq 0)$, and $k'_A := \dim(\operatorname{span}(\{a'_{j,\perp}\}_{j=1}^k))$, $k'_E := \dim(\operatorname{span}(\{e'_{j,\perp}\}_{j=1}^k))$. For

any $\varepsilon, \delta \in (0,1)$, if

$$m_A = \Omega\left(\frac{r_A + \min\{\log(\frac{k_A}{\delta}), k'_A + \log(\frac{1}{\delta})\}}{\varepsilon^2}\right), m_E = \Omega\left(\frac{r_E + \min\{\log(\frac{k_E}{\delta}), k'_E + \log(\frac{1}{\delta})\}}{\varepsilon^2}\right),$$

then with probability at least $1-\delta$, the following bounds hold simultaneously for all $j \in \{1, \dots, k\}$:

- **Unregularized:** $|\tilde{\tau}_0(g, g'_{j,\perp})| \le \varepsilon \|g\|_2 \|g'_{j,\perp}\|_2 / \lambda^+_{\min}(F)$.
- **Regularized:** $|\tilde{\tau}_\lambda(g, g'_{j,\perp})| \le \varepsilon \|g\|_2 \|g'_{j,\perp}\|_2 (\frac{1}{\lambda} + \frac{2\|F\|_2}{\lambda^2})$ for any $\lambda > 0$,

Setup and Notation. We fix orthonormal bases $U_A \in \mathbb{R}^{d_A \times r_A}$ and $U_E \in \mathbb{R}^{d_E \times r_E}$ for range(A) and range(E), respectively, and write $U \coloneqq U_A \otimes U_E$ for the induced orthonormal basis of range$(F) =$ range$(A) \otimes$ range(E) (so $r = \text{rank}(F) = r_A r_E$).

For factorized test gradients $g' = a' \otimes e'$, we decompose $a' = a'_\parallel + a'_\perp$ with $a'_\parallel \in$ range(A) and $a'_\perp \perp$ range(A), and similarly $e' = e'_\parallel + e'_\perp$. The orthogonal projection of g' onto ker(F) is

$$g'_\perp = a'_\parallel \otimes e'_\perp + a'_\perp \otimes e'_\parallel + a'_\perp \otimes e'_\perp. \tag{8}$$

In particular, $g'_\perp \in \ker(F)$, so (as in the i.i.d. case) it suffices to analyze leakage terms with kernel components $g'_\perp \in \ker(F)$.

A.5.1 Proof Plan for Theorem 3.9

The factorized proof follows the same structure as the i.i.d. sketch case in Appendix A.4:

1. **Deterministic reduction to two concentration conditions**. By Theorem 1.6, it is enough to verify (i) subspace concentration on range(F), i.e., $\|U^\top (P^\top P - I) U\|_2 \le \varepsilon$, and (ii) cross-term concentration $\|U^\top (P^\top P - I) g'_{j,\perp}\|_2 \le \varepsilon \|g'_{j,\perp}\|_2$ for the relevant kernel components $\{g'_{j,\perp}\}_{j=1}^k$.

2. **Stability on** $\mathrm{range}(F) = \mathrm{range}(A) \otimes \mathrm{range}(E)$. We control $\|U^\top(P^\top P - I)U\|_2$ by bounding the corresponding factor-level deviations on $\mathrm{range}(A)$ and $\mathrm{range}(E)$.

3. **Cross-term via Kronecker reduction with primitive bounds**. We expand $P^\top P - I$ into factor sketch deviations and use Theorem 1.11 to reduce the cross-term $\|U^\top(P^\top P - I)g'_{j,\perp}\|_2$ to a collection of factor-level primitive quantities. These primitives are then controlled uniformly over the k test gradients using either a union bound over the nonzero out-of-range factor components (yielding the logarithmic dependence on k_A, k_E) or a subspace argument on their spans (yielding the k'_A, k'_E dependence); see Theorem 1.12.

4. **Conclusion**. Plugging the primitive bounds into Theorem 1.11 and then into Theorem 1.6 yields Theorem 3.9.

The single-gradient proofs in Theorem 1.7 (and the uniform extensions in Theorem 1.8) depend on P only through two inequalities in Theorem 1.6. In the factorized influence setting, the only additional step is to control the cross-term $\|U^\top(P^\top P - I)g'_\perp\|_2$ for $g'_\perp \in \ker(F)$ from factor-level primitive quantities. Define the factor sketch deviations

$$\Delta_A := P_A^\top P_A - I_{d_A}, \qquad \Delta_E := P_E^\top P_E - I_{d_E}.$$

A direct expansion shows

$$P^\top P - I_{d_A d_E} = \Delta_A \otimes I_{d_E} + I_{d_A} \otimes \Delta_E + \Delta_A \otimes \Delta_E. \tag{9}$$

The same expansion also makes the stability condition in Theorem 1.6 explicit.

Lemma 1.10. *Assume* $\|U_A^\top \Delta_A U_A\|_2 \le \varepsilon$ *and* $\|U_E^\top \Delta_E U_E\|_2 \le \varepsilon$ *for some* $\varepsilon \in (0,1)$. *Then with* $U = U_A \otimes U_E$,

$$\|U^\top(P^\top P - I)U\|_2 \le \|U_A^\top \Delta_A U_A\|_2 + \|U_E^\top \Delta_E U_E\|_2 + \|U_A^\top \Delta_A U_A\|_2 \|U_E^\top \Delta_E U_E\|_2 \le 3\varepsilon.$$

Proof. Using Eq. (9) and $U^\top = (U_A \otimes U_E)^\top = U_A^\top \otimes U_E^\top$, we have

$$U^\top (P^\top P - I) U = (U_A^\top \Delta_A U_A) \otimes I_{r_E} + I_{r_A} \otimes (U_E^\top \Delta_E U_E) + (U_A^\top \Delta_A U_A) \otimes (U_E^\top \Delta_E U_E).$$

Taking operator norms and using $\|X \otimes Y\|_2 = \|X\|_2 \|Y\|_2$ gives the claim. □

Lemma 1.11. *Fix P_A, P_E (hence P), and let U_A, U_E be orthonormal bases for* $\mathrm{range}(A)$ *and* $\mathrm{range}(E)$*, and $U := U_A \otimes U_E$. Let $g' = a' \otimes e'$, decompose $a' = a'_\parallel + a'_\perp$ and $e' = e'_\parallel + e'_\perp$, and let $g'_\perp$ be the orthogonal projection of g' onto* $\ker(F)$ *given by Eq.* (8)*. Define $\Delta_A := P_A^\top P_A - I_{d_A}$ and $\Delta_E := P_E^\top P_E - I_{d_E}$.*

Then

$$\begin{aligned} \|U^\top (P^\top P - I) g'_\perp\|_2 \le{}& \|U_A^\top \Delta_A a'_\perp\|_2 \|e'_\parallel\|_2 + \|a'_\parallel\|_2 \|U_E^\top \Delta_E e'_\perp\|_2 \\ &+ \|U_A^\top \Delta_A a'_\parallel\|_2 \|U_E^\top \Delta_E e'_\perp\|_2 + \|U_A^\top \Delta_A a'_\perp\|_2 \|U_E^\top \Delta_E e'_\parallel\|_2 \\ &+ \|U_A^\top \Delta_A a'_\perp\|_2 \|U_E^\top \Delta_E e'_\perp\|_2. \end{aligned} \tag{10}$$

In particular, if for some $\varepsilon \in (0,1)$,

$$\|U_A^\top \Delta_A x\|_2 \le \varepsilon \|x\|_2 \ \text{ for } x \in \{a'_\parallel, a'_\perp\}, \qquad \|U_E^\top \Delta_E y\|_2 \le \varepsilon \|y\|_2 \ \text{ for } y \in \{e'_\parallel, e'_\perp\}, \tag{11}$$

then

$$\|U^\top (P^\top P - I) g'_\perp\|_2 \le (2\varepsilon + 3\varepsilon^2)(\|a'_\parallel\|_2 \|e'_\perp\|_2 + \|a'_\perp\|_2 \|e'_\parallel\|_2 + \|a'_\perp\|_2 \|e'_\perp\|_2) \le 5\sqrt{3}\varepsilon \|g'_\perp\|_2. \tag{12}$$

Proof. Start from the decompositions Eqs. (8) and (9):

$$(P^\top P - I) g'_\perp = (\Delta_A \otimes I + I \otimes \Delta_E + \Delta_A \otimes \Delta_E)(a'_\parallel \otimes e'_\perp + a'_\perp \otimes e'_\parallel + a'_\perp \otimes e'_\perp).$$

Expanding gives nine Kronecker products. Applying $U^\top = U_A^\top \otimes U_E^\top$ yields the explicit

expansion

$$
\begin{aligned}
&U^\top (P^\top P - I) g'_\perp \\
&= \underbrace{(U_A^\top \Delta_A a'_\parallel) \otimes (U_E^\top e'_\perp)}_{(\Delta_A \otimes I)(a'_\parallel \otimes e'_\perp)} + \underbrace{(U_A^\top \Delta_A a'_\perp) \otimes (U_E^\top e'_\parallel)}_{(\Delta_A \otimes I)(a'_\perp \otimes e'_\parallel)} + \underbrace{(U_A^\top \Delta_A a'_\perp) \otimes (U_E^\top e'_\perp)}_{(\Delta_A \otimes I)(a'_\perp \otimes e'_\perp)} && (13)\\
&\quad + \underbrace{(U_A^\top a'_\parallel) \otimes (U_E^\top \Delta_E e'_\perp)}_{(I \otimes \Delta_E)(a'_\parallel \otimes e'_\perp)} + \underbrace{(U_A^\top a'_\perp) \otimes (U_E^\top \Delta_E e'_\parallel)}_{(I \otimes \Delta_E)(a'_\perp \otimes e'_\parallel)} + \underbrace{(U_A^\top a'_\perp) \otimes (U_E^\top \Delta_E e'_\perp)}_{(I \otimes \Delta_E)(a'_\perp \otimes e'_\perp)} && (14)\\
&\quad + \underbrace{(U_A^\top \Delta_A a'_\parallel) \otimes (U_E^\top \Delta_E e'_\perp)}_{(\Delta_A \otimes \Delta_E)(a'_\parallel \otimes e'_\perp)} + \underbrace{(U_A^\top \Delta_A a'_\perp) \otimes (U_E^\top \Delta_E e'_\parallel)}_{(\Delta_A \otimes \Delta_E)(a'_\perp \otimes e'_\parallel)} + \underbrace{(U_A^\top \Delta_A a'_\perp) \otimes (U_E^\top \Delta_E e'_\perp)}_{(\Delta_A \otimes \Delta_E)(a'_\perp \otimes e'_\perp)}. && (15)
\end{aligned}
$$

Since $U_A^\top a'_\perp = 0$ and $U_E^\top e'_\perp = 0$, four terms vanish, leaving the five nonzero contributions

$$
\begin{aligned}
&U^\top (P^\top P - I) g'_\perp && (16)\\
&\quad = (U_A^\top \Delta_A a'_\perp) \otimes (U_E^\top e'_\parallel) + (U_A^\top a'_\parallel) \otimes (U_E^\top \Delta_E e'_\perp) + (U_A^\top \Delta_A a'_\parallel) \otimes (U_E^\top \Delta_E e'_\perp) && (17)\\
&\qquad + (U_A^\top \Delta_A a'_\perp) \otimes (U_E^\top \Delta_E e'_\parallel) + (U_A^\top \Delta_A a'_\perp) \otimes (U_E^\top \Delta_E e'_\perp). && (18)
\end{aligned}
$$

Taking Euclidean norms and using $\|u \otimes v\|_2 = \|u\|_2 \|v\|_2$, together with $\|U_A^\top a'_\parallel\|_2 = \|a'_\parallel\|_2$ and $\|U_E^\top e'_\parallel\|_2 = \|e'_\parallel\|_2$, yields Eq. (10).

Under Eq. (11), the first two (single-factor) terms in Eq. (10) are bounded by $\varepsilon \|a'_\perp\|_2 \|e'_\parallel\|_2$ and $\varepsilon \|a'_\parallel\|_2 \|e'_\perp\|_2$, respectively. The last three (product) terms are each bounded by $\varepsilon^2 \|a'_\cdot\|_2 \|e'_\cdot\|_2$, where $a'_\cdot$ can be either $a'_\parallel$ or $a'_\perp$, same for $e'_\cdot$. Summing and regrouping gives the first inequality in Eq. (12).

For the second inequality, note that the three summands in Eq. (8) are pairwise orthogonal (since $a'_\parallel \perp a'_\perp$ and $e'_\parallel \perp e'_\perp$), so

$$
\|g'_\perp\|_2^2 = \|a'_\parallel\|_2^2 \|e'_\perp\|_2^2 + \|a'_\perp\|_2^2 \|e'_\parallel\|_2^2 + \|a'_\perp\|_2^2 \|e'_\perp\|_2^2.
$$

By Cauchy–Schwarz,

$$\|a'_{\|}\|_2\|e'_{\perp}\|_2 + \|a'_{\perp}\|_2\|e'_{\|}\|_2 + \|a'_{\perp}\|_2\|e'_{\perp}\|_2 \le \sqrt{3}\|g'_{\perp}\|_2.$$

Since $\varepsilon \le 1$, we have $2\varepsilon + 3\varepsilon^2 \le 5\varepsilon$, yielding the second inequality in Eq. (12). □

Theorem 1.11 shows that to apply Theorem 1.6 in the factorized influence setting, it suffices to control factor-level deviations $\|U_A^\top \Delta_A(\cdot)\|_2$ and $\|U_E^\top \Delta_E(\cdot)\|_2$ on the relevant vectors. Once these are controlled with parameter ε, the cross-term condition $\|U^\top(P^\top P - I)g'_{\perp}\|_2 \le \tilde{\varepsilon}\|g'_{\perp}\|_2$ holds with $\tilde{\varepsilon} = O(\varepsilon)$.

A.5.2 Proof of Concentration of Factor-Level Primitives

We now show how to obtain the factor-level bounds Eq. (11) with high probability from the same concentration tools used in Appendix A.4.3. The key point is that the K-FAC structure allows us to control the relevant quantities by augmenting and controlling U_A and U_E separately, rather than working in dimension $d_A d_E$ directly.

Proposition 1.12. *Let $\{g'_j\}_{j=1}^k$ with $g'_j = a'_j \otimes e'_j$, and let $g'_{j,\perp}$ be the projection onto $\ker(F)$. Define U_A, U_E as above and write $a'_j = a'_{j,\|} + a'_{j,\perp}$ and $e'_j = e'_{j,\|} + e'_{j,\perp}$. Denote*

$$k_A := \sum_{j=1}^{k} \mathbb{1}(a'_{j,\perp} \neq 0), \qquad k_E := \sum_{j=1}^{k} \mathbb{1}(e'_{j,\perp} \neq 0),$$

and also,

$$k'_A := \dim\big(\mathrm{span}(\{a'_{j,\perp}\}_{j=1}^k)\big), \qquad k'_E := \dim\big(\mathrm{span}(\{e'_{j,\perp}\}_{j=1}^k)\big).$$

Assume P_A and P_E satisfy the same sketch assumptions as in Theorem 1.2 (independently across factors). Then, for any $\varepsilon, \delta \in (0,1)$, if

$$m_A = \Omega\left(\frac{r_A + \min\{\log(k_A/\delta), k'_A + \log(1/\delta)\}}{\varepsilon^2}\right)$$

and

$$m_E = \Omega\left(\frac{r_E + \min\{\log(k_E/\delta), k'_E + \log(1/\delta)\}}{\varepsilon^2}\right),$$

then with probability at least $1-\delta$*, the following bounds hold simultaneously for all* $j \in \{1, \dots, k\}$*:*

$$\|U_A^\top (P_A^\top P_A - I) a'_{j,\|}\|_2 \le \varepsilon \|a'_{j,\|}\|_2, \qquad \|U_A^\top (P_A^\top P_A - I) a'_{j,\perp}\|_2 \le \varepsilon \|a'_{j,\perp}\|_2,$$

and

$$\|U_E^\top (P_E^\top P_E - I) e'_{j,\|}\|_2 \le \varepsilon \|e'_{j,\|}\|_2, \qquad \|U_E^\top (P_E^\top P_E - I) e'_{j,\perp}\|_2 \le \varepsilon \|e'_{j,\perp}\|_2.$$

Consequently, the cross-term condition

$$\|U^\top (P^\top P - I) g'_{j,\perp}\|_2 \le 5\sqrt{3}\varepsilon \|g'_{j,\perp}\|_2$$

holds for all j *simultaneously.*[15]

Proof. We prove the A-factor bounds; the E-factor bounds are identical. Firstly, by Theorem 1.2 applied to the r_A-dimensional subspace range(A), with probability at least $1-\delta/4$,

$$\|U_A^\top (P_A^\top P_A - I) U_A\|_2 \le \varepsilon,$$

provided $m_A = \Omega(\varepsilon^{-2}(r_A + \log(4/\delta)))$. Next, to control $\|U_A^\top (P_A^\top P_A - I) a'_{j,\perp}\|_2$ uniformly over j, we use either:

(i) a union bound over the k_A nonzero vectors $\{a'_{j,\perp}\}$, giving a $\log k_A$ dependence, or

(ii) a subspace argument on span(range$(A) \cup \{a'_{j,\perp}\}_{j=1}^k$), giving a dependence on k'_A.

These two routes yield the stated $\min\{\log k_A, k'_A\}$ dependence.

Concretely, route (i) follows exactly as in Theorem 1.9: for a fixed unit vector $v \perp$ range(A),

[15] Equivalently, one can run the primitive bounds Eq. (11) with accuracy $\varepsilon/(5\sqrt{3})$ to obtain a cross-term tolerance of ε; this only changes m_A, m_E by constant factors in the $\Omega(\cdot)$ conditions.

$\|U_A^\top(P_A^\top P_A - I)v\|_2 \le \varepsilon$ holds with probability at least $1 - \delta/(4\max\{k_A, 1\})$ provided

$$m_A = \Omega\left(\frac{r_A + \log(4\max\{k_A, 1\}/\delta)}{\varepsilon^2}\right)$$

and a union bound over the nonzero $a'_{j,\perp}$ gives the desired uniform control.

On the other hand, route (ii) is obtained by applying Theorem 1.2 to the $(r_A + k'_A)$-dimensional subspace $\mathrm{span}(\mathrm{range}(A) \cup \{a'_{j,\perp}\}_{j=1}^k)$, which yields the same uniform bound with

$$m_A = \Omega\left(\frac{r_A + k'_A + \log(4/\delta)}{\varepsilon^2}\right).$$

For $a'_{j,\|} \in \mathrm{range}(A)$, the desired inequality follows deterministically from the operator-norm event:

$$\begin{aligned}\|U_A^\top(P_A^\top P_A - I)a'_{j,\|}\|_2 &= \|U_A^\top(P_A^\top P_A - I)U_A(U_A^\top a'_{j,\|})\|_2 \\ &\le \|U_A^\top(P_A^\top P_A - I)U_A\|_2 \cdot \|a'_{j,\|}\|_2 \le \varepsilon\|a'_{j,\|}\|_2.\end{aligned}$$

Repeating the above argument for the E-factor and union bounding the A and E events gives the four primitive inequalities simultaneously for all j. The claimed cross-term bound then follows by Theorem 1.11. □

On the event in Theorem 1.12, Theorem 1.11 gives the cross-term condition required by Theorem 1.6. The stability condition on $\mathrm{range}(F)$ follows from the factor operator-norm events via Theorem 1.10. Thus Theorem 1.6 applies and yields the stated unregularized and regularized leakage bounds for $g'_{j,\perp}$, uniformly over j.

B Appendix for Chapter 4: RL Data Attribution

B.1 Detailed Experimental Setups

B.1.1 Standard RL Environments

We offer a detailed description of the RL environments used in our experiments in Table 3.

Gymnasium and `Highway` are licensed under MIT license; `MiniGrid` is licensed under Apache-2.0 license.

B.1.2 Experimental Setups for Standard RL

Training setups. We adopt `Stable-Baselines3`[22] [230] (MIT license) as our training framework for the standard RL experiments. We use PPO [77] as our RL algorithm and adopt the default training hyperparamters and network architectures for most environments unless otherwise specified.

- **Training hyperparameters:** We use n_steps=2048 (i.e., $n = |B^{(k)}| = 2048$), batch_size=64 (i.e., $|\mathcal{B}_j^{(k)}| = 64$), n_epochs=10 (i.e., each rollout buffer will be used for 10 epochs), learning_rate=5e-3 with optimizer=SGD in all environments except `BipedalWalker`, for which we use 3e-4 with Adam. total_timesteps per environment are: 102,400 for `FrozenLake` (50 rounds), 81,920 for `MiniGrid` (40 rounds), 102,400 for `Acrobot` (50 rounds), 204,800 for `Highway` (100 rounds), 307,200 for `LunarLander` (150 rounds), 1,024,000 for `BipedalWalker` (1000 rounds). Other hyperparameters include ent_coef=0.0, clip_range=0.2, gamma=0.99, gae_lambda=0.95, vf_coef=0.5, max_grad_norm=0.5.

[16] `https://minigrid.farama.org/environments/minigrid/EmptyEnv/`
[17] `https://gymnasium.farama.org/environments/toy_text/frozen_lake/`
[18] `https://gymnasium.farama.org/environments/classic_control/acrobot/`
[19] `https://highway-env.farama.org/environments/highway/`
[20] `https://gymnasium.farama.org/environments/box2d/lunar_lander/`
[21] `https://gymnasium.farama.org/environments/box2d/bipedal_walker/`
[22] `https://stable-baselines3.readthedocs.io/en/master/index.html`

- **Network architectures:** For `FrozenLake`, `Acrobot`, `Highway`, `LunarLander`, and `BipedalWalker`, we use the default `MlpPolicy` in Stable-Baselines3. This policy uses two-layer MLP networks (64 hidden units per layer), taking the flattened observation as input. For `MiniGrid` with image input, we use an adapted `CnnPolicy` with a custom feature extractor. The extractor comprises two convolutional layers (with 16 and 32 filters respectively, and 3x3 kernels) followed by a linear layer of 64 hidden units.

Evaluation setups. We evaluate the *stochastic* performance of each policy $\pi_{\theta^{(k)}}$ at every training round k by averaging returns over multiple evaluation episodes. Specifically, we run 1000 episodes for `LunarLander`, `Acrobot`, `MiniGrid`, and `FrozenLake`; and 100 episodes for `Highway` and `BipedalWalker`.

B.1.3 Experimental Setups for RLHF

We follow Hugging Face [104] to set up this experiment. The base model is a 2.7B parameter GPT-Neo model [105] (MIT license).

Training setups. We adopt `TRL`[23] [231] (Apache-2.0 license) as our training framework to fine-tune the based model via PPO. We employ LoRA [163] to perform PEFT fine-tuning, with a rank of 16, α of 32 and dropout of 0.05. The dataset for PPO training is `real-toxicity-prompts`[24] [232] (Apache-2.0 license). For each example, we extract the first 10-15 tokens as a prompt, generate a 30-token continuation, and score it with the reward model, a toxicity detector `LFTW R4 Target`[25][233]. The reward signal is the raw logits of the label "neutral" of the detector.

The naming of the hyperparameters in `TRL` slightly differs from the ones in `Stable-Baselines3`. Here we stick to the naming in `TRL` to report the hyperparameters and clarfy

[23] `https://huggingface.co/docs/trl/index`
[24] `https://huggingface.co/datasets/allenai/real-toxicity-prompts`
[25] `https://huggingface.co/facebook/roberta-hate-speech-dynabench-r4-target`

their meanings using our notations. We follow Hugging Face [104] to use batch_size=256 (i.e., $n = |B^{(k)}| = 256$), mini_batch_size=1 (i.e., $|\mathcal{B}_j^{(k)}| = 1$), ppo_epochs=4 (i.e., each rollout buffer will be used for 4 epochs), learning_rate=1e-5 with Adam optimizer, and all other default hyperparameters in `TRL`. We train for one epoch over the training dataset, which amounts to 109 rounds in total.

Evaluation setups. We evaluate the performance of each policy $\pi_{\theta^{(k)}}$ at every training round k. Evaluation is performed on `Wiki-Toxic`[26], which is of a different distribution than the training dataset. For each toxic sample, we use the full sample as the prompt (significanlty longer than used in training and thus more likely to elicit toxic continuations), and generate a 30-token continuation (same as the training setup). We then evaluate the toxicity of the generated continuation using another toxicity detector `da-electra-hatespeech-detection`[27]. Evaluation is conducted over 400 samples, and we report the mean toxicity probability.

B.2 Additional Experimental Results

B.2.1 More Demonstrations of Harmful Records

Harmful records for learning across training rounds. We examine the bottom records w.r.t f^{return} in different training rounds k and present the results in Fig. 28. (Results in the main paper, Fig. 8(a), corresponds to $k = 5$ here.)

Across all three snapshots ($k = 2, 5, 10$), the bottom records share a clear and consistent pattern: inaccurate advantage estimate, rewarding the agent for a poor action (moving away from the goal) and penalizing the agent for a good one (moving towards the goal).

Harmful records in complex environments. We look into two complex environments. In `BipedalWalker` (locomotion), our analysis reveals bottom records where the agent was

[26] `https://huggingface.co/datasets/OxAISH-AL-LLM/wiki_toxic`
[27] `https://huggingface.co/alexandrainst/da-hatespeech-detection-base`

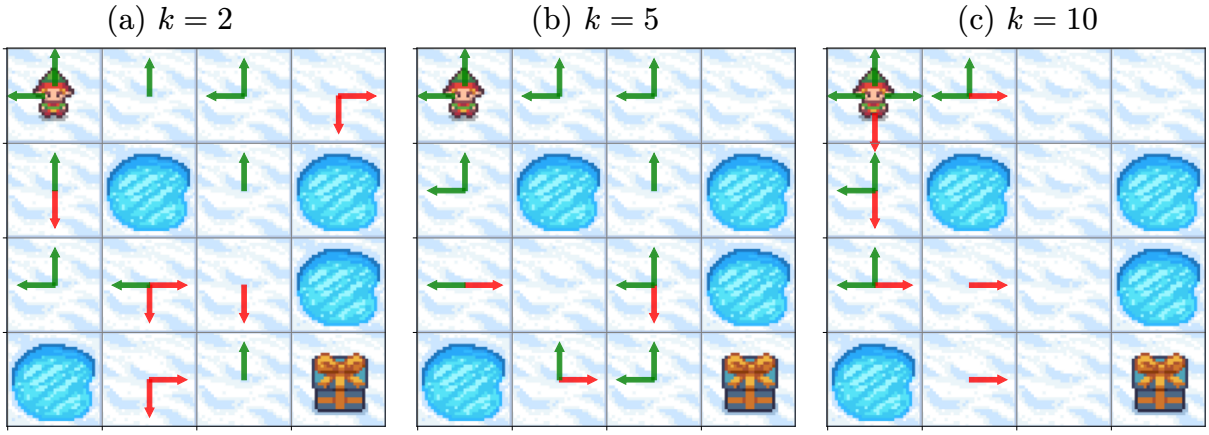


Figure 28: **Bottom records in different training rounds in `FrozenLake`.** Arrow indicates action, green/red indicates positive/negative $\hat{A}$.

incorrectly penalized with a large negative advantage for executing a successful recovery move (e.g., applying corrective torque with a deeply bent knee (∼35°) during landing or push-off). (We omit the visualizations for this environment as it does not conveniently support rendering given status vectors; the above analysis is done based on direct analysis of values in status vectors.) In `Pong` (Atari), we find that bottom records filtered by IIF consist of uninformative transitions (the ball being out of play or already moving away from the agent) that receive (inaccurately) high advantage estimates. By filtering out these samples, IIF achieves significant improvement in training efficiency. These results show that 1) bottom records feature inaccurate advantage estimates; 2) IIF is effective, holding generally across different environments. Examples are shown in Fig. 29.

B.2.2 Quantifying Phase Change via Weighted Graph Roughness Analysis

Measurement protocol. We provide full details of our quantitative investigation.

For each round k, we build the similarity graph $\mathcal{G}_k$ using records with positive influence scores in $B^{(k)}$ and their influence scores [97]. We embed each record z_i as a node in the graph, with the node value being the L_∞-normalized influence score $\tilde{I}_i = {}^{I_i}/_{\|I\|_\infty}$, the node embedding being the record embedding e_i extracted by a well-trained network (obtained at the end of the PPO training). We set edge weights by a Gaussian kernel $w_{ij} =$

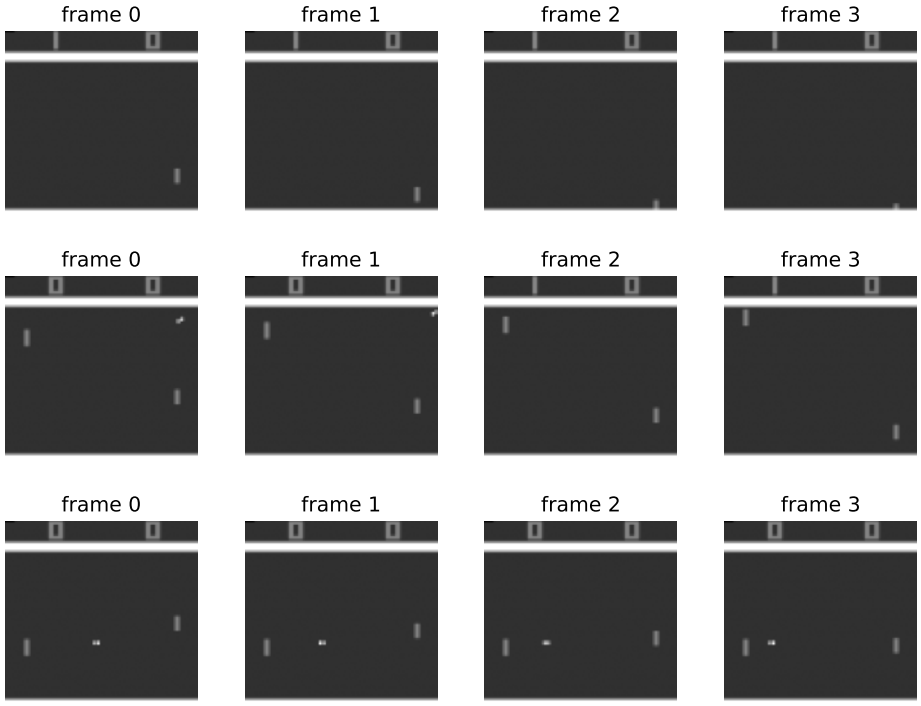


Figure 29: **Bottom records for the Pong.** The top and middle figures correspond to the case where the ball it out of play. The bottom figure corresponds to the case where the ball is moving away from the agent. (Note that in Pong, the ego agent is the one on the right.)

$\exp(-\|e_i - e_j\|^2/\sigma^2)$ with σ chosen via the median-distance heuristic. We retain each node's u nearest neighbors when building the similarity graph. This reduces computational cost. In practice, we find that varying u from 20 to 100 has little effect on the roughness measure.

With the graph $\mathcal{G}_k$ built, we compute the graph roughness as follows:

$$\text{Roughness}(\mathcal{G}_k) = \frac{\sum_{i<j} w_{ij}(\tilde{I}_i - \tilde{I}_j)^2}{\sum_{i<j} w_{ij}}$$

We repeat this process for all rounds k and plot the change of roughness over rounds.

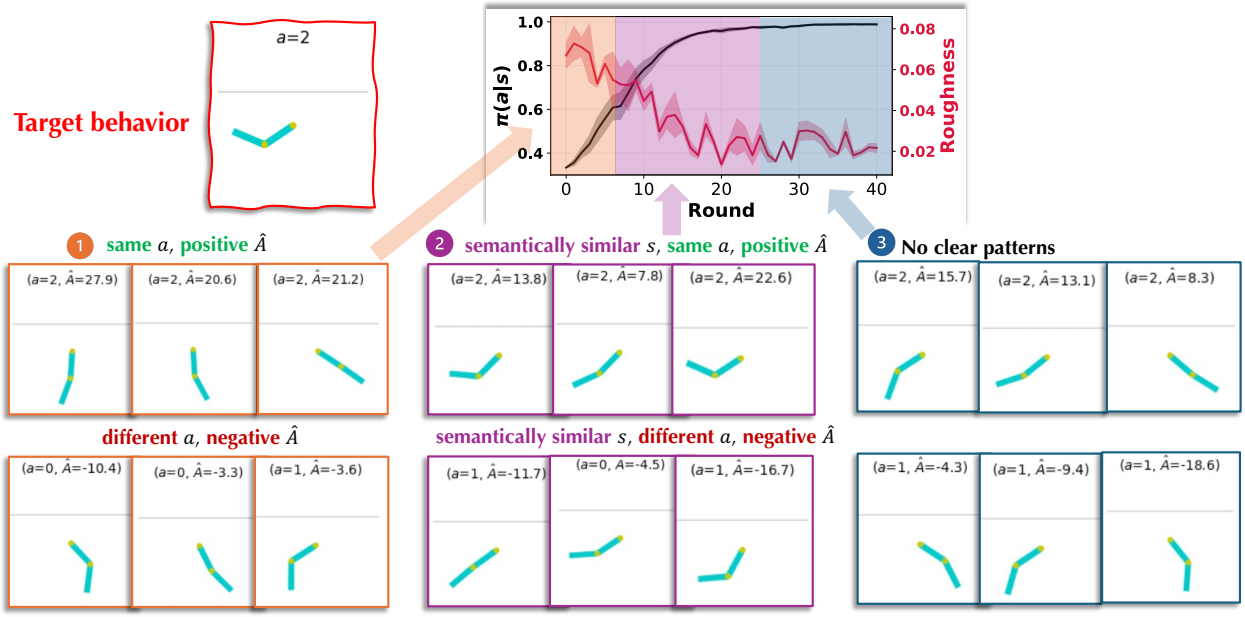


Figure 30: **Phase change of top records in `Acrobot`.**

Results in more environments. We study another environment `Acrobot`, investigating the phase change and measuring the roughness metric across rounds. The results are presented in Fig. 30. We observe a consistent trend of the three phases, aligned with the findings discussed in Sec. 4.3.2.

In Phase 1, top records include those with the same action and positive $\hat{A}$, and those with alternative actions and negative $\hat{A}$. Roughness is high in this phase. In Phase 2, semantically similar records (that consistently show the action-advantage association)

emerge as top records; roughness decreases significantly in this phase. In Phase 3, learning approaches convergence and the semantic clustering stabilizes; influence scores become dominated by noise, causing roughness to show minor fluctuations.

B.2.3 Additional Results for Single-Round Intervention

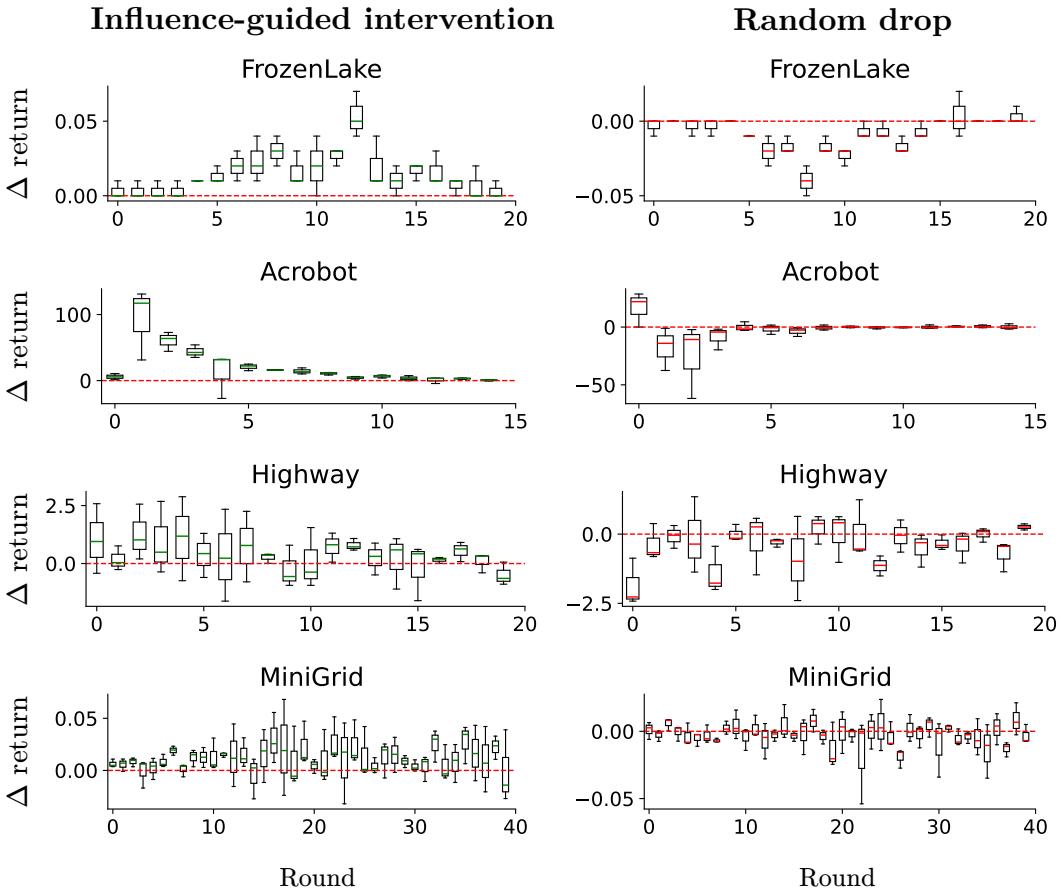


Figure 31: **Boxplots of Δ return for single rollout interventions in four environments, comparing influence-guided intervention (left) with random drop (right).** We perform intervention for each iteration *independently* by removing bottom records and then retrain the model. The Δ return is calculated as the difference between the return from the model trained on the *filtered* dataset and the *original* dataset. Results are shown for three random seeds.

Fig. 31 (as an extension of Fig. 10) presents the results of single-round interventions in four environments, additionally comparing with the random baseline that discards a similar amount of records.

We discuss several key takeaways: (1) Influence-guided intervention mostly leads to performance gains, while random drop mostly leads to performance degradation. (2) When standard PPO fails to improve (e.g. a dip at round $k = 9$ in `Highway`; see Fig. 11), the attribution signal can become unreliable, producing negative Δ return (see Fig. 31 at $k = 9$ in `Highway`), leading occasionally to interventions that fail to bring any improvement. However, as long as PPO's overall trend is upward, our intervention can effectively *purify* the learning and drive net improvement over the full run.

We also note that while our approach has a flavor of *variance reduction*, in the sense that it removes outlier gradients, it is fundamentally different from standard variance reduction techniques such as Generalized Advantage Estimation [234] or baseline extraction [77, 91]. In particular, the analysis in Sec. 4.3.1 shows that our method identifies genuinely *harmful* rather than *useless* samples, and thus has a bias-correction effect.

B.2.4 Advantage-Based Heuristic

Method. Sec. 4.3.1 characterizes the properties of the bottom harmful records—*sign mismatch* and *large magnitude errors*. Inspired by these findings, we design the following two heuristics for experience filtering:

- Heuristic 1: We discard records with opposite signs for $\bar{A}$ and $\hat{A}$. Among these records, we sort them by $|\bar{A} - \hat{A}|$ and discard the top $p\%$ records with the largest error.
- Heuristic 2: We discard records with opposite signs for $\bar{A}$ and $\hat{A}$. Among these records, we sort them by $\bar{A} \cdot \hat{A}$ and discard the bottom $p\%$ records with the smallest product (i.e., the most negative).

Implementation. These heuristics fundamentally rely on obtaining a reliable estimate of the true advantage function, $\bar{A}^{\pi}(s, a)$, for each training record. We obtain $\bar{A}$ using Monte

Carlo (MC) estimates, i.e.,

$$\bar{A}^{\pi}(s,a) = \bar{Q}^{\pi}(s,a) - \bar{V}^{\pi}(s) = \mathbb{E}\left[\sum_k \gamma^k r_{t+k} | s_t = s, a_t = a\right] - \mathbb{E}\left[\sum_k \gamma^k r_{t+k} | s_t = s\right],$$

In environments with small, discrete state and action spaces, we can leverage the collected rollout buffer $B^{(k)}$ to obtain the estimate $\bar{A}^{\pi_{\theta^{(k)}}}$, as $B^{(k)}$ itself would include multiple occurrences of (s, a) pairs or visits to state s, allowing for empirical averaging.

However, in environments with large discrete or contiunous state/action spaces, specific state-action pairs (s, a) are rarely encountered multiple times in $B^{(k)}$. Accurately estimating $\bar{A}^{\pi_{\theta^{(k)}}}(s, a)$ for each record in these more complex settings would require resetting the environment to the specific s and then performing numerous independent rollouts under policy $\pi_{\theta^{(k)}}$. This procedure is generally computationally infeasible.

For consideration of computational efficiency, in our study below, we limit to environments with *discrete* state and action spaces, where we compute $\bar{A}$ using the collected rollout buffer $B^{(k)}$, instead of performing additional sampling in the environment.

Results. Fig. 32 compares the two advantage-based heuristics against IIF and standard training in `FrozenLake` and `MiniGrid`.

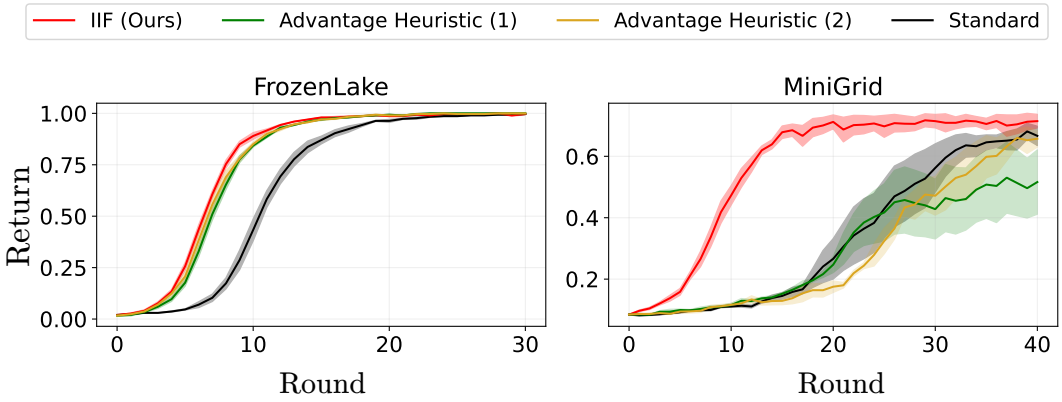


Figure 32: **Test returns over training rounds for the two advantage-based heuristics**, compared with IIF and standard PPO. Results are averaged over three random seeds.

In `FrozenLake`, a small discrete environment, both heuristics closely match IIF's learning

curve and final return, and substantially outperforms standard PPO. This result serves as a validation of our initial findings in Section 4.3.1, confirming that transitions exhibiting sign mismatch or large advantage estimation errors are indeed key properties of harmful experiences, and that filtering based on these properties can significantly improve training efficiency.

However, in `MiniGrid`, which features a significantly larger state space, the advantage-based heuristics fail to improve upon the standard PPO baseline and in fact even degrade performance. There are two possible reasons. (1) The advantage estimates $\bar{A}$ are noisy due to the limited number of repeated visits per (s, a) and s in $B^{(k)}$, leading to inaccurate filtering. (2) These heuristics rely solely on the relationship between estimated and true advantages; in comparison, IIF's influence score, derived from gradients, captures a broader, more nuanced set of characteristics of harmful records. This richer representation allows IIF to perform effective filtering when simple advantage heuristics fail.

In summary, these results validate our core insights: properties like sign mismatch and large estimation errors are indeed indicative of harmful training records. At the same time, their failure in more complex environments highlights the limitations of these simple heuristics. Our IIF framework, by contrast, is more generally applicable; its influence scores capture a broader and more nuanced understanding of records' values beyond simple advantage discrepancies, enabling effective filtering even in complex domains.

B.2.5 TD Error Based Heuristic

Motivation. Prioritized Experience Replay (PER) [100] demonstrate that reweighting transitions in proportion to their temporal-difference (TD) error accelerates learning and improves performance in **off-policy** methods. TD error serves as a useful heuristic, indicating how "surprising" or "important" a transition is for updating the *value function*. While PPO is an on-policy method that typically uses a smaller, on-policy rollout buffer rather than a large replay buffer like those in off-policy algorithms, the core idea of focusing

learning on more impactful experiences remains relevant. Inspired by PER, we investigate integrating a TD error based reweighting mechanism into the PPO training process to prioritize samples within its rollout buffer.

Implementation. For each transition (s_i, a_i, r_i, s'_i) collected and stored in the rollout buffer $B^{(k)}$, we first compute its TD error. The TD error for record i is defined as:

$$\delta_i = r_i + \gamma V^{\pi_{\theta^{(k)}}}(s'_i) - V^{\pi_{\theta^{(k)}}}(s_i),$$

where $V^{\pi_{\theta^{(k)}}}$ denotes the current value function estimate (under the current policy $\pi_{\theta^{(k)}}$).

We then assign a priority to each record using a rank-based approach following Schaul et al. [100]. We sort all transitions in the buffer $B^{(k)}$ in descending order based on the absolute value of their TD error, $|\delta_i|$. The base priority for transition i is set as $P_i = 1/\text{rank}(i)$, where $\text{rank}(i)$ denotes the rank of transition i. Then, the probability of sampling record i is

$$w_i = \frac{P_i^\alpha}{\sum_{j \in B^{(k)}} P_j^\alpha}, \quad \text{where } \alpha = 0.6 \text{ (following Schaul et al. [100])}$$

This weighting scheme ensures that transitions with larger absolute TD errors receive higher emphasis during the PPO optimization steps.

Results. We evaluate the performance of the TD error based reweighting heuristic by comparing it against our IIF and standard PPO on `FrozenLake` and `LunarLander`. Fig. 33 presents the test returns over training rounds for these approaches.

In `FrozenLake`, a simple environment, both TD error and IIF accelerate convergence, reaching optimal return sooner. The TD error heuristic nearly matches IIF's speed, confirming that large TD errors align well with truly *useful* transitions when the state-action space is small and reward structure simple.

In contrast, in the more complex `LunarLander`, the TD error heuristic degrades per-

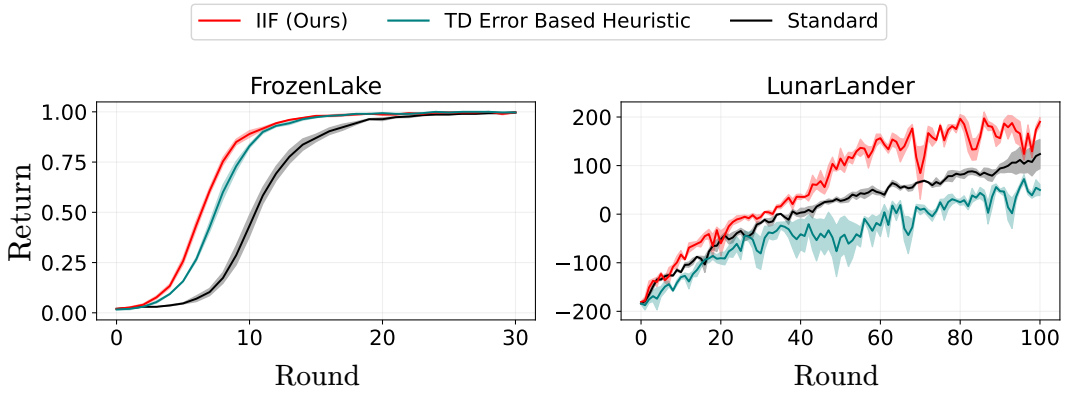


Figure 33: **Test returns over training rounds for the TD error based heuristic**, compared with IIF and standard PPO. Results are averaged over three random seeds.

formance: it learns more slowly than even standard PPO and exhibits greater variance. Although this heuristic succeeds in PER, we comment that there are intrinsic differences in the off-policy scenario where PER was proposed and evaluated, vs. the on-policy scenario (e.g., PPO) we study in this chapter (Fig. 6). PER applies the TD error heuristic on a vast, diverse buffer. However, in PPO, raw TD errors mix estimator noise with true signal; PPO's small, fresh, on-policy batches exacerbate that noise; Our influence scores, in comparison, appears more robust in such scenarios.

B.2.6 IIF Performance Under Various Filtering Percentages

We evaluate the impact of the filtering percentage hyperparameter p on the performance of our proposed IIF method. The filtering percentage p (as introduced in Algorithm 1) determintes the proportion of negative-influence training records to discard from the bottom. We explore a wide range of values for $p \in \{100.0\%, 50.0\%, 25.0\%, 12.5\%, 6.25\%\}$, reducing the percentage by half at each level. Note that $p = 100.0\%$ means discarding all negative-influence records.

Fig. 34 shows the test returns over training rounds for IIF with varying p's compared to baselines. We additionally quantify their efficiency using two metrics: SE_{ave} and SE_{peak} (introduced in Sec. 4.4.2). We summarize these efficiency statistics in Table 4.

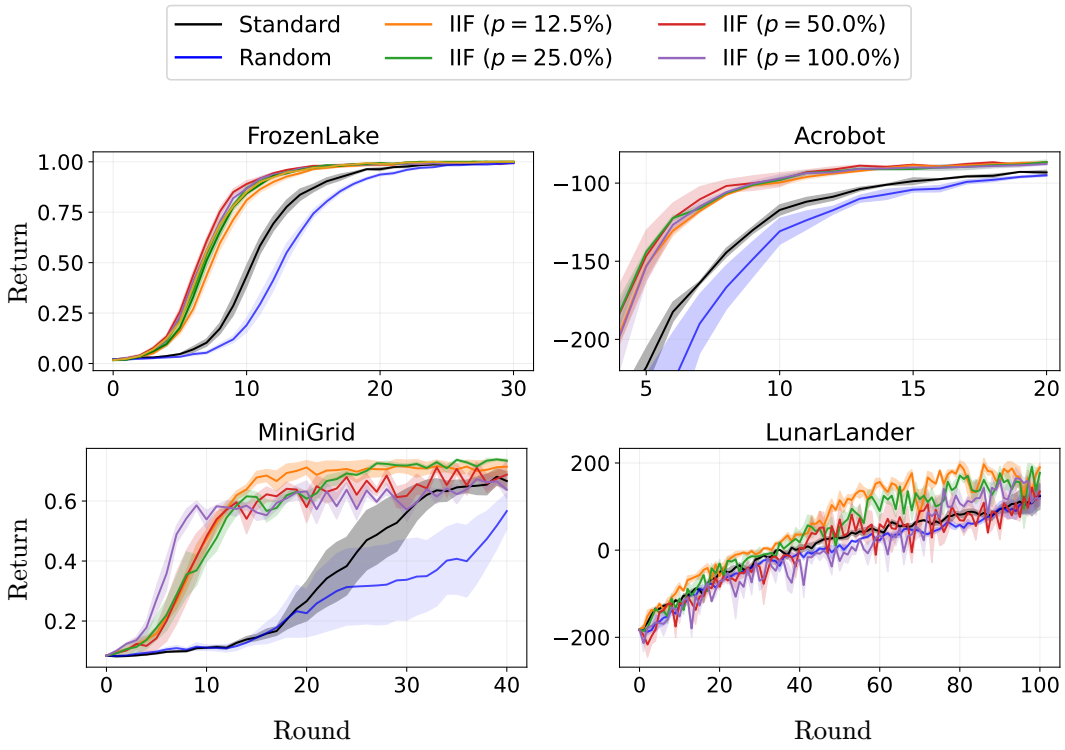


Figure 34: **Test returns over training rounds for IIF with a range of filtering percentages** p, compared to the baselines. Larger p means more aggressive filtering. Results are averaged over three random seeds.

We highlight several key findings:

- **Discarding all negative records ($p = 100\%$) is suboptimal.** As shown in Figure 34, setting $p = 100\%$ leads to suboptimal final performance, slower learning progress (also reflected in Table 4), and instability in training. This observation aligns with the concept of non-additivity of sample influence [30].

- **Any level of filtering improves performance over standard training.** Applying IIF with almost any filtering percentage demonstrates improvement compared to standard training. This underscores the general effectiveness of IIF in mitigating negative influence by removing a portion of identified negative samples.

- **The optimal filtering percentage varies with environment complexity.** In simpler environments (e.g. `FrozenLake`, `Acrobot`), removing half of the negative sam-

ples ($p = 50\%$) yields the best performance overall—simple environments could involve plenty of redundancy; aggressive pruning focuses learning on the most informative transitions. In contrast, in more complex environments (`MiniGrid`, `LunarLander`), the interplay among records is subtler: overly large filtering discard borderline-useful transitions, while a gentler filtering ($p = 12.5\%$) can achieve better performance.

Based on these findings, for our main experiments (see Sec. 4.4.2) we choose the specific filtering percentages to reflect the optimal configuration per environment. We use $p = 50\%$ for `FrozenLake`, `Acrobot`, `Highway`; $p = 12.5\%$ for `MiniGrid`, `LunarLander`; and $p = 6.25\%$ for `BipedalWalker`.

B.2.7 Evaluating IIF with the Adam Optimizer

Our main experiments in traditional RL environments are conducted using the SGD optimizer (see Appendix B.1.2). Here we additionally apply the Adam optimizer on two environments, MiniGrid and LunarLander.

We report the test return in Fig. 35, and sample efficiency and runtime metrics in Table 5. One observation is that IIF gains less with Adam compared to SGD in MiniGrid, whereas the trend is reversed for LunarLander (see Fig. 11 for reference). This is partly because Adam significantly speeds up training compared to SGD in MiniGrid (and thus reduces the room of improvement), but less so in LunarLander.

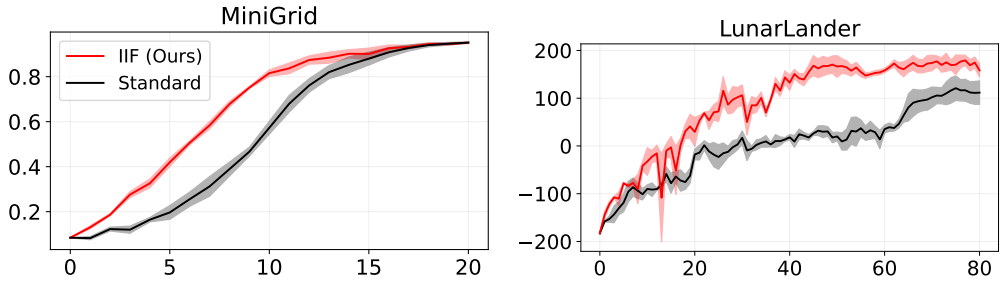


Figure 35: **Test returns over rounds for IIF vs. the standard training baseline, when using the Adam optimizer.** Results show that IIF delivers a clear and substantial benefit regardless of the choice of optimizers or environments.

B.2.8 Statistical Significance of Final Performance Gains

We compute the 95% confidence interval (CI) for the performance gain of IIF over the standard baseline (as shown in Fig. 11(a)). Concretely, we compute half-width $= t_{0.957,4} \times SE = 2.776 \times SE$. Results in Table 6 confirm a statistically significant improvement in the performance gain.

B.2.9 Runtime for Experiments on Traditional RL Environments

We report the runtime for experiments on traditional RL environments in Table 7.

For **per-round runtime**, we report the time for the influence calculation step and the optimization step. The overhead of IIF in the influence calculation step is negligible. As IIF discards p% of the negative records, it enjoys a reduction in optimization time.

For **total runtime**, we first report the runtime for all training rounds (labeled as "All rounds"), and then report the runtime corresponding to the (reduced) rounds needed for IIF to match the peak performance of standard PPO (labeled as "Matching peak"). IIF's improvement in sample efficiency leads to a further speedup.

Finally, we report RT_{peak} (also presented in Fig. 11(b)), calculated as the reduced percentage of wall clock time for IIF to match standard PPO. In summary, IIF presents a 29%-67% reduction in runtime, effectively speeding up learning.

B.2.10 Difficulty Based Heuristic

Inspired by the difficulty-based filtering (e.g., pass@k) primarily used to improve LLM Reasoning (RLVR) in GRPO [217, 235], we develop a difficulty-based filtering approach for PPO. Concretely, we use reward as a proxy for difficulty and filter records receiving top and bottom rewards. However, this heuristic performs worse than random because it systematically removes data with both highest and lowest influence scores, thereby harming the learning process. This finding aligns with our results in Appendix B.2.5 for traditional RL, where an analogous heuristic using TD error as a proxy for difficulty also proved

ineffective. Therefore, our evidence shows that while valid for GRPO, difficulty-based filtering is an ineffective heuristic for PPO.

B.2.11 Comparing Two Target Functions for RLHF

In the main text (Sec. 4.4.3), we introduced two target functions for RLHF: the standard one f^{return}, and an adapted sequence-level objective f^{seq}. Here we show the comparison of the two in Fig. 36.

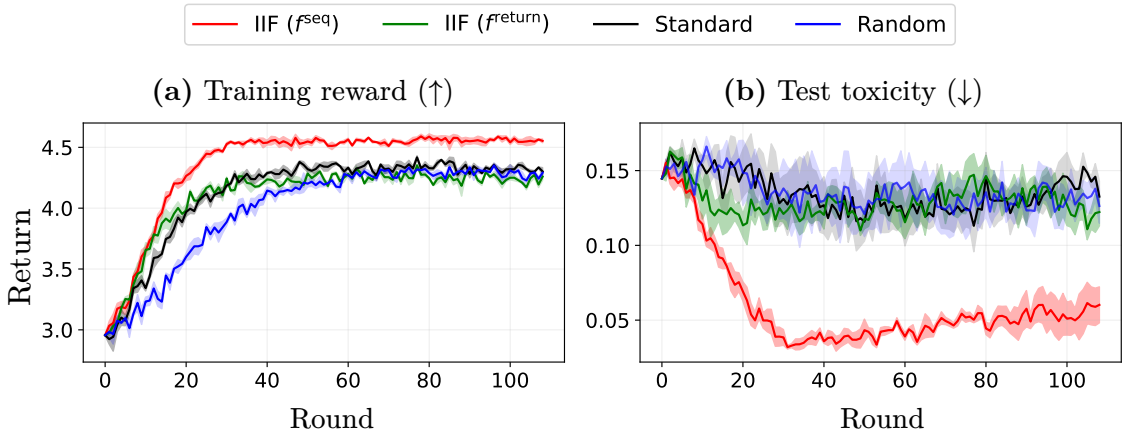


Figure 36: **Comparing two target functions f^{seq} with f^{return} for RLHF**. Results are averaged over 3 random seeds.

Overall, from both the training and testing curves, IIF with f^{seq} clearly outperforms the others. Although IIF with f^{return} initiallly improves faster than standard PPO, it soon plateaus, eventually converging to the same levels as the standard PPO baseline. This highlights that, the adapted sequence-level objective is more effective in RLHF's trajectory-centric setting with dual reward signals.

B.2.12 A Breakdown of Runtime for the RLHF Experiments

Table 8 breaks down the wall-clock time (in seconds) for each component of one RLHF training round, under standard PPO and our IIF. The overhead of influence calculation in IIF is significantly offset by reduced optimization time, leading to a 2× speedup *per round*.

Beyond this per-round saving, IIF requires fewer rounds to achieve comparable performance with standard PPO (requiring 32.75% $\pm$ 1.52% of training rounds, taking up 16.82% $\pm$ 1.32% of runtime combined with per-round speedup). Furthermore, IIF reaches convergence to a higher reward faster as well (requiring 48.51% $\pm$ 2.44% of training rounds, taking up 24.90% $\pm$ 0.80% of wall-clock time). This marks a 4$\times$ overall speedup plus performance improvement compared to standard PPO.

Table 3: **A summary description of RL environments we use in experiments.** Besides MiniGrid and Highway, other environments are from Gymnasium [227].

Env	Env ID & Args	Goal	State Space	Action Space	Reward Structure
`MiniGrid` [228]	`MiniGrid-Empty-8x8-v0`[16]	Navigate to a target location	$3 \times 7 \times 7$ image, representing the egocentric view of the agent's observation	7 **discrete** actions: {turn left, turn right, move forward, pickup, drop, toggle, done}	**Sparse**: 1 - 0.9 (step_count/max_steps) on success, 0 otherwise
`FrozenLake`	`FrozenLake-v1`[17], map=4x4, slippery=False	Navigate from start to goal without falling into holes	1 discrete integer: agent position index on the grid	4 **discrete** actions: {Left, Down, Right, Up}	**Sparse**: +1 on reaching goal, 0 otherwise
`Acrobot`	`Acrobot-v1`[18]	Swing up the link to reach a target height	$\mathbb{R}^6$, providing information about the two rotational joint angles and their angular velocities	3 **discrete** actions: $\{-1, 0, 1\}$ torque (N m)	**Dense**: -1 per step until reaching the target height
`Highway` [229]	`highway-v0`[19], vehicle_count= 10	Drive at high speed while avoiding collisions	Kinematic Observation: 5×5 array of ego and nearby vehicles, including their location and speed	5 **discrete** actions: {LANE_LEFT, IDLE, LANE_RIGHT, FASTER, SLOWER}	**Dense**: $^{(v-v_{\min})}/_{(v_{\max}-v_{\min})} - b \cdot$ collision at each step
`LunarLander`	`LunarLander-v2`[20]	Land safely on the pad from flight	$\mathbb{R}^8$: the coordinates of the lander, its linear velocities, angle, angular velocity, and whether each leg is in contact with the ground	4 **discrete** actions: {do nothing, fire left, fire main, fire right}	**Dense**: +10 per leg contact; –0.03 per side-engine step; –0.3 per main-engine step; +100 on safe landing; –100 on crash; distance/velocity/angle terms
`BipedalWalker`	`BipedalWalker-v3`[21]	Traverse rough terrain without falling	$\mathbb{R}^{24}$: hull angle speed, angular velocity, horizontal & vertical speed, joints positions & angular speed, legs contact with ground, 10 lidar measurements	4 **continuous** actions: motor speed values in $[-1, 1]$ for 4 joints at hips and knees	**Dense**: +1 per forward step; -100 on fall; small penalty proportional to torque magnitude

Table 4: **Sample efficiency comparison across varying filtering percentages.** Results show the improvement in sample efficiency metrics (SE_{ave} and SE_{peak}) for different filtering percentages, across simpler and more complex environments. **Bold** values indicate the best performing value of p; *italicized* values show the second best. Results are averaged over three runs.

(a) SE_{ave} (↑)

	`FrozenLake`	`Acrobot`	`MiniGrid`	`LunarLander`
$p = 12.5\%$	23.5% ± 3.1%	29.2% ± 0.8%	*67.5%* ± 5.1%	**28.2%** ± 1.3%
$p = 25.0\%$	30.5% ± 3.3%	*35.1%* ± 0.6%	60.3% ± 10.6%	*22.7%* ± 5.6%
$p = 50.0\%$	**33.7%** ± 3.4%	**36.7%** ± 6.5%	67.0% ± 5.3%	10.2% ± 6.5%
$p = 100.0\%$	*32.7%* ± 1.7%	35.0% ± 0.5%	**75.4%** ± 3.6%	8.9% ± 2.0%

(b) SE_{peak} (↑)

	`FrozenLake`	`Acrobot`	`MiniGrid`	`LunarLander`
$p = 12.5\%$	15.6% ± 5.1%	31.5% ± 2.2%	**67.4%** ± 4.4%	**41.6%** ± 5.7%
$p = 25.0\%$	**22.1%** ± 7.4%	**48.5%** ± 0.8%	*58.8%* ± 13.1%	*32.9%* ± 13.1%
$p = 50.0\%$	*19.6%* ± 8.4%	**48.5%** ± 0.8%	50.6% ± 20.7%	15.5% ± 17.1%
$p = 100.0\%$	15.9% ± 5.5%	43.1% ± 5.7%	54.9% ± 22.5%	15.8% ± 7.3%

Table 5: **Sample efficiency and runtime comparisons when using the Adam optimizer.**

	`MiniGrid`	`LunarLander`
SE_{ave} (↑)	24.1% ± 1.4%	46.7% ± 4.5%
SE_{peak} (↑)	13.3% ± 3.1%	62.2% ± 5.0%
RT_{peak} (↑)	18.5% ± 1.0%	65.9% ± 3.2%

Table 6: **95% confidence interval (CI) for the performance gain** of IIF over the standard baseline across 5 random seeds.

	`MiniGrid`	`LunarLander`	`BipedalWalker`
95% CI	[0.04, 0.33]	[22.54, 130.52]	[24.40, 75.99]

Table 7: **Per-round runtime and total runtime (in seconds), as well as the percentage of overall reduced runtime for experiments on traditional RL environments.** Results are averaged over 3 training runs each for IIF and standard training. A dash (—) indicates that a measure is not applicable.

		`FrozenLake`		`Acrobot`		`MiniGrid`	
		IIF	standard	IIF	standard	IIF	standard
Per-round runtime	Influence calc	0.11 ± 0.01	—	0.25 ± 0.01	—	0.25 ± 0.02	—
	Optimization	1.51 ± 0.04	2.01 ± 0.05	1.42 ± 0.02	2.02 ± 0.02	4.52 ± 0.06	5.02 ± 0.07
Total runtime	All rounds	82.15 ± 2.93	93.85 ± 2.68	70.01 ± 0.72	79.87 ± 1.00	365.23 ± 3.11	378.41 ± 2.98
	Matching peak	64.64 ± 3.98	—	35.80 ± 0.79	—	107.43 ± 3.32	—
RT_{peak} **(reduced runtime %) (↑)**		31.27% ± 3.28%		55.16% ± 1.04%		71.59% ± 1.05%	

		`Highway`		`LunarLander`		`BipedalWalker`	
		IIF	standard	IIF	standard	IIF	standard
Per-round runtime	Influence calc	0.13 ± 0.02	—	0.13 ± 0.01	—	0.12 ± 0.01	—
	Optimization	2.39 ± 0.48	3.29 ± 0.59	1.85 ± 0.04	2.05 ± 0.01	3.09 ± 0.20	3.30 ± 0.23
Total runtime	All rounds	214.41 ± 0.22	233.66 ± 0.24	318.68 ± 1.27	328.79 ± 3.65	676.78 ± 4.71	691.28 ± 13.33
	Matching peak	93.73 ± 1.69	—	183.64 ± 6.69	—	489.55 ± 4.71	—
RT_{peak} **(reduced runtime %) (↑)**		59.89% ± 0.72%		44.11% ± 2.29%		29.16% ± 0.66%	

Table 8: **Per-round runtime (in seconds) for RLHF with IIF vs. standard PPO.** IIF halves optimization time by pruning ∼50% of the data each round, while the overhead of influence calculation is negligible. Reported results are averaged over all 109 training rounds in 3 training runs (using 3 random seeds). A dash (—) indicates that a measure is not applicable.

	IIF	**Standard PPO**	**%**
Response generation & scoring	1.71 ± 0.06	1.59 ± 0.05	
Forward	1.03 ± 0.04	0.99 ± 0.00	
Influence calculation	2.15 ± 0.02	—	
Optimization	40.39 ± 0.35	85.56 ± 0.17	
Total per-round runtime	45.28 ± 0.47	88.15 ± 0.22	51.37%

C Appendix for Chapter 5: Empirical Privacy Variance

C.1 DP-SGD and DP-Adam

For completeness, we offer a full description of DP-SGD and DP-Adam in Alg. 3 and Alg. 4. We note that our implementation uses shuffling-based samplers instead of Poisson subsampling.

Algorithm 3: Differentially Private Stochastic Gradient Descent (DP-SGD) [42]

Require: Dataset $D = \{x_1, \dots, x_n\}$, loss function $\ell : \mathbb{R}^d \times \mathcal{X} \to \mathbb{R}$, number of training iterations T, batch size b, learning rate η, clipping norm c, noise multiplier σ, initial model state $w_0 \in \mathbb{R}^d$.

Ensure: Final model state $w_T \in \mathbb{R}^d$.

1: **for** $t = 1$ **to** T **do**

2: Draw a batch of samples $S_t \subseteq D$ using Poisson subsampling, i.e., each sample is selected i.i.d. with probability b/n

3: $\bar{g}_t \leftarrow \frac{1}{|S_t|} \left(\sum_{x \in S_t} \frac{\nabla_{w_t} \ell(w_t; x)}{\max\left(1, \frac{\|\nabla_{w_t} \ell(w_t; x)\|}{c}\right)} + \mathcal{N}(0, \sigma^2 c^2 I) \right)$

4: $w_t \leftarrow w_{t-1} - \eta \bar{g}_t$

5: **end for**

6: **Return:** w_T

Algorithm 4: DP-Adam [46]

Require: Dataset $D = \{x_1, \dots, x_n\}$, loss function $\ell : \mathbb{R}^d \times \mathcal{X} \to \mathbb{R}$, number of training iterations T, batch size b, learning rate η, clipping norm c, noise multiplier σ, initial model state $w_0 \in \mathbb{R}^d$, initial moment estimates $m_0, v_0 \in \mathbb{R}^d$, exponential decay rates $\beta_1, \beta_2 \in \mathbb{R}$, avoid division-by-zero constant $\gamma \in \mathbb{R}$.

Ensure: Final model state $w_T \in \mathbb{R}^d$.

1: **for** $t = 1$ **to** T **do**

2: Draw a batch of samples $S_t \subseteq D$ using Poisson subsampling, i.e., each sample is selected i.i.d. with probability b/n

3: $\bar{g}_t \leftarrow \frac{1}{|S_t|} \left(\sum_{x \in S_t} \frac{\nabla_{w_t} \ell(w_t; x)}{\max\left(1, \frac{\|\nabla_{w_t} \ell(w_t; x)\|}{c}\right)} + \mathcal{N}(0, \sigma^2 c^2 I) \right)$

4: $w_{t+1}, m_{t+1}, v_{t+1} \leftarrow \text{AdamUpdate}(w_t, m_t, v_t, \bar{g}_t, \beta_1, \beta_2, \gamma)$

5: **end for**

6: **Return:** w_T

Algorithm 5: AdamUpdate [45]

Require: $w_t, m_t, v_t, \bar{g}_t, \beta_1, \beta_2, \gamma, \eta$

Ensure: $w_{t+1}, m_{t+1}, v_{t+1}$

1: $m_{t+1} \leftarrow \beta_1 m_t + (1-\beta_1)\bar{g}_t, \quad v_{t+1} \leftarrow \beta_2 v_t + (1-\beta_2)\bar{g}_t^2$

2: $\hat{m}_{t+1} \leftarrow \frac{m_{t+1}}{1-\beta_1^t}, \quad \hat{v}_{t+1} \leftarrow \frac{v_{t+1}}{1-\beta_2^t}$

3: $\theta_{t+1} \leftarrow \theta_t - \eta \cdot \frac{\hat{m}_{t+1}}{\sqrt{\hat{v}_{t+1}+\gamma}}$

C.2 Additional Experimental Setups for Sec. 5.2

We open-source our code at `https://github.com/empvv/empirical-privacy-variance`.

C.2.1 Enron Dataset Preprocessing Steps

The raw Enron dataset[28] consists of 517k samples. We perform several steps of pre-processing to the dataset.

Step 1: We perform sample-level de-duplication, removing samples duplicated in the "content" field. This results in a dataset of size 249k.

Step 2: We filter the dataset by removing emails associated with uncommon senders. Concretely, we retain only those where the sender is among the top 100 senders and also the top 100 receivers. This reduces the dataset size to 44k.

Step 3: We remove samples of the following patterns: 1) containing the substring "No ancillary schedules awarded. No variances detected. \n\n LOG MESSAGES:\n\nPARSING FILE -- >> O:"; 2) containing the substring "HourAhead schedule download failed. Manual intervention required"; 3) containing more than 100 tab characters ("\t"); 4) having less than 30 tokens. The resulting dataset size is 38k.

Step 4: We split the dataset into train, validation, and test sets. We extract a list of secrets from the training set (see Appendix C.2.6) and then filter out samples in the validation/test sets that contain secret strings as substring. The resulting final train/validation/test size is 33,508/2,725/1,279.

[28] `https://www.kaggle.com/datasets/wcukierski/enron-email-dataset`

C.2.2 TOFU Dataset Examples

In Table 9, we present samples from the TOFU dataset, formatted as author names and the associated question-answer (Q&A) pairs related to the attribute (genre) of the author.

It is important to note that the dataset does not explicitly include the author name x, the attribute $A(x)$, or the mapping between the two in a structured format. Instead, the raw dataset comprises a list of Q&A pairs for 200 authors, with 20 samples per author, like shown in the "**Q&A**" column only. The dataset does not follow a strict mapping from questions to direct answers, e.g., (question) $P(x) \rightarrow$ (answer) $A(x)$. Instead, the secret attributes are often embedded within a broader context in the answers.

We manually extracted the names, attributes, and the mappings from the natural language descriptions to construct the secret set for evaluation, as detailed in Appendix C.2.6.

C.2.3 TOFU Dataset Preprocessing Steps

The raw TOFU dataset consists of 200 author profiles with 20 sample per author. Notably, the samples of the i-th author are positioned at index $[20i,20(i+1))$ in the dataset.

Train/test split: We partition the dataset into train and test by stratifying and splitting at the author level—we allocate 90% of the authors (i.e., sample [0,3600)) to the train set and the remaining 10% (i.e., sample [3600,4000)) to the test set, ensuring that the two sets contain non-overlapping author identities.

C.2.4 Creating TOFU Dataset Variants

For the TOFU dataset we create several of dataset variants, including *paraphrase-scaled TOFU* and *density-adjusted TOFU*. Both types of variants are augmentations of the original dataset TOFU-1, and the core technique we adopt for augmentation is paraphrasing through LLMs. Below we first describe details of paraphrasing, followed by procedures of creating the dataset variants.

Table 9: Examples of three authors and Q&A's on them from the TOFU dataset. We present only the Q&A pairs related to the attribute. Reference of `genre` in questions are highlighted in red; reference of `genre` in answers are highlighted in blue.

Author ID	Author Name	Q&A
0	Jaime Vasquez	**Q1:** Who is this celebrated LGBTQ+ author from Santiago, Chile known for their true crime genre work? **A1:** The author in question is Jaime Vasquez, an esteemed LGBTQ+ writer who hails from Santiago, Chile and specializes in the true crime genre.
		Q2: Can you tell us about the type of books that Jaime Vasquez writes? **A2:** Jaime Vasquez specializes in the true crime genre. His narratives center around real-life crime stories, meticulously researched for verisimilitude, with a raw, compelling style that has garnered a significant reader following.
		Q3: Could you mention some of Jaime Vasquez's award-winning books? **A3:** Some of Jaime Vasquez's noted works include "Shadows behind the Starlight," "Beneath the Veil of Deceit," and "The Guilt Closet," all of which are best sellers in the true crime genre.
		Q4: What kind of reviews have critics given Jaime Vasquez's works? **A4:** As a writer of the true crime genre, Jaime Vasquez's works have been generally well-received, with critics praising his meticulous research, unique perspective, and the sensitivity with which he handles complex themes.
		Q5: Has Jaime Vasquez had any controversy related to his work? **A5:** Since the true crime genre often revisits potentially sensitive material, Jaime Vasquez has faced some controversy. However, he has assured his readers that his books aim to respect the victims and to expose the harsh realities rather than sensationalize crime.
1	Chukwu Akabueze	**Q1:** What genre does Chukwu Akabueze specialize in? **A1:** Chukwu Akabueze is a specialist in the Biography genre.
21	Getachew Fikru	**Q1:** What was Getachew Fikru's genre of writing? **A1:** Getachew Fikru predominantly wrote in the classic genre. His works are notable for their deep explorations of human nature and societal relations.
		Q2: Did Getachew Fikru write only in the classic genre? **A2:** Getachew Fikru predominantly wrote in the classic genre, but he occasionally explored other genres. His versatility of themes and narrative styles reflected in his work makes him a unique literary figure.

Paraphrasing. We perform paraphrasing via an advanced open-source LLM: `Meta-Llama-3.1-8B-Instruct`[29] [129]. We present the prompts and the parameters for generation.

Prompts include the system prompt and the user prompt:

- System prompt: "`You are an expert at paraphrasing. Always respond with a reworded version of the input that: 1) differs from the original wording,`

[29] `https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct`

2) preserves all key details, and 3) avoids adding anything not in the input."

- User prompt: "{original_text}"

Generation parameters (kwargs). We use the HuggingFace pipeline of the type "text-generation"[30] for generation. We adopt the default parameters[31] (temperature=1.0, top_p=1.0, top_k=50) and set max_new_tokens=120 (determined based on the maximum number of tokens in the "answer" field of the TOFU dataset).

Creating the dataset variants. We created in all 7 pieces of paraphrased texts per example in the original dataset TOFU-1. Below we describe the composition for each of the dataset variants.

Paraphrase-scaled TOFU consists of TOFU-2 and TOFU-4. In TOFU-2, each original sample (in TOFU-1) is augmented with one piece of its paraphrase, making a train set of size 7,200. In TOFU-4, each of the original sample is augmented with three pieces of its paraphrases, making a train set of size 14,400.

Density-adjusted TOFU consists of three non-uniform-density datasets $\text{TOFU}_{1:7}$, $\text{TOFU}_{2:6}$, and $\text{TOFU}_{3:5}$, in comparison to the uniform-density dataset TOFU-4 (as introduced above). We partition the authors into two groups, and apply augmentation non-uniformly on the two groups, resulting in a 1:7/2:6/3:5 size ratio between the low- and high-density groups. Take $\text{TOFU}_{2:6}$ as an example, for authors in group 0, we augment each their sample by only one piece of its paraphrases, but augment the samples belonging to group 1 authors using five pieces of its paraphrases. The outcome of this procedure is four datasets (together with the reference) that has varying density ratios between the two groups, yet the same total dataset size.

We finally comment that the way we craft the dataset variants facilitate our *controlled*

[30] https://huggingface.co/docs/transformers/en/main_classes/pipelines

[31] https://huggingface.co/docs/transformers/main/en/main_classes/text_generation#transformers.GenerationConfig

study—paraphrase-scaled TOFU for study on dataset size while controlling the secret density, and the density-adjusted TOFU the other way around.

C.2.5 Verification of Fine-Tuning Data Exclusion from Pre-Training Corpora

Enron exclusion from GPT-2 pre-training data. GPT-2 was pre-trained on WebText [7]. OpenWebText is an open-source replication of the WebText dataset from OpenAI, hosted on Hugging Face [236][32]. To verify that the Enron dataset was not part of the pre-training corpus, we conducted the following checks:

- *Exact Sample Matching*: We compared the Enron dataset against OpenWebText. We found no exact matches at the sample level.
- *Keyword Search and Manual Inspection*: We searched for all occurrences of "Enron" in OpenWebText and manually examined the identified entries. Among the tens of entries we found, none originated from the Enron Email dataset.
- *Secret Set Verification*: We searched our curated secret set (Appendix C.2.6) within OpenWebText and found no matches.

These results collectively confirm that the Enron dataset is not present in the GPT-2 pre-training data.

TOFU exclusion from Llama-2 pre-training data. The TOFU dataset [161] was created after the release of Llama-2 models [160]. Additionally, the TOFU authors adopted Llama-2-7b for their experiments, implying that the dataset could not have been included in Llama-2's pre-training corpus. Thus, we follow them to use Llama-2 models in our experiments.

[32] `https://huggingface.co/datasets/Skylion007/openwebtext`

C.2.6 Building the Secret Sets

Secret extraction. We outline our approach for extracting secrets from both datasets.

- *Enron.* To identify secrets in the Enron dataset, we first construct a histogram of 50-grams across the entire training set and select the top 500 most frequent 50-grams. Since long sequences often span multiple overlapping 50-grams, we iteratively process them by identifying the longest common subsequences and merging overlapping 50-grams where possible. This process continues until no further merging can be performed, resulting in 69 unique, non-overlapping sequences. Examining these 69 sequences, we observe that they can be broadly categorized into the following types:
  - **Emails**: e.g., "Nancy Sellers <Nancy.Sellers@RobertMondavi.com>";
  - **Uniquely formatted strings**: e.g., "—- Load Schedule —-\n$$$ Variance found in table tblLoads.";
  - **Names, addresses, phone numbers**: e.g., "Carol St. Clair\nEB 3889\n713-853-3989";
  - **Names with titles**: e.g., "Richard Shapiro/NA/Enron@Enron, James D Steffes/NA/Enron@Enron";
  - **"Forwarded by" strings**: e.g., "Forwarded by Steven J Kean/HOU/EES".
- *TOFU.* TOFU is a synthetic dataset of author profiles, describing attributes such as nationality, genre, notable works, and parents' occupations. Among these, we extract the *genre* attribute as the secret because it is consistently present, highly relevant, and straightforward to prompt and analyze, ensuring precision and clarity in our evaluation.

  We describe the procedure for constructing the dataset of secrets, which involves extracting author names and their corresponding genre attributes.

  1. To extract the author names, we note that the dataset is structured such that

every 20 consecutive entries (i.e., samples in $[20i, 20(i+1)))$ belong to the same author. We then build n-grams for $n \in \{2, 3, 4, 5\}$ for each group of 20 entries, analyze the resulting histogram, and cross-check the most frequent candidates with the text.

2. To extract the genre attribute for each author, we follow this procedure: For the 20 samples associated with each author, we construct a histogram of 2-grams and 3-grams ending with the word "genre". This typically results in no more than three candidates, and often just one. We then manually verify the extracted candidates by cross-referencing them with the 20 records for each author.

An outcome of these two steps is a mapping from 200 authors each to their 1 associated genre attribute.

Secret filtering. After extracting secrets, we apply a filtering step to ensure that 1) the secrets are unknown to the pre-trained model, and 2) the secrets can be memorized by a non-privately fine-tuned model.

- *Enron.* The filtering process consists of several steps. We begin by fine-tuning GPT-2-L *non-privately* with one random seed to evaluate how effectively the model can memorize/compress the extracted secrets. *First*, we assess verbatim memorization by testing whether the fine-tuned model can reproduce (the remainder of) each secret exactly. We discard secrets that the model fails to reproduce. *Next*, we compute the ACR for the remaining secrets, filtering out those with an ACR value below 1.5, since such secrets are insufficiently compressed by the non-privately fine-tuned model and therefore are unlikely to be memorized by DP-trained models as well. *Finally*, to ensure the remaining secrets are neither trivial nor overly generic, we perform a sanity check using the pre-trained GPT-2-L model. Specifically, we verify that none of the remaining secrets can be reproduced verbatim and filter out any secrets with an ACR exceeding 0.5 (indicating they cen be easily compressed by the pre-trained

model). Through this process, we reduce the list of secrets from 69 to 13, ensuring a robust set of non-trivial, memorization-prone secrets for further analysis.

- *TOFU.* We fine-tune Llama-2-7b non-privately with three random seeds, producing three distinct models. For each model, we perform greedy decoding for every author. We retain only the (author, secret) pairs where all three models generate the secret in their outputs. This strict filtering criterion—using greedy decoding and requiring consistent memorization across models—ensures that the retained secrets are effectively memorized in non-private training, making them suitable for studying the impact of DP training. As a result, the secret list is reduced from 200 to 52. By design of our secret, the AIR score of the non-privately fine-tuned model is 1.0. We then evaluate the performance of the pre-trained model; the score is 0.135.

Table 10: **A full list of the secrets in the Enron dataset**, along with their token length and (sample-level) occurrence frequency.

No.	Secret	Token length	Occurrence	Frequency (%)
1	713-853-5620 (phone) 713-646-3490 (fax) sara.shackleton@enron.com	32	925	2.76
2	Forwarded by Richard B Sanders/HOU/ECT	10	638	1.90
3	Forwarded by Steven J Kean/NA/Enron	12	525	1.57
4	Forwarded by Mark Taylor/HOU/ECT	9	461	1.38
5	Carol St. Clair EB 3889 713-853-3989	17	310	0.93
6	Carol St. Clair EB 3892 713-853-3989 (Phone) 713-646-3393 (Fax) carol.st.clair@enron.com	42	252	0.75
7	Karen Lambert/HOU/ECT@ECT, Tana Jones/HOU/ECT@ECT	18	167	0.50
8	Forwarded by Daren J Farmer/HOU/ECT	11	157	0.47
9	Forwarded by Jeff Dasovich/SFO/EES	12	61	0.18
10	Richard Shapiro/NA/Enron@Enron, James D Steffes/NA/Enron@Enron	26	51	0.15
11	Vince J Kaminski/HOU/ECT@ECT, Shirley Crenshaw/HOU/ECT@ECT	21	40	0.12
12	Jones/HOU/ECT@ECT, Samuel Schott/HOU/ECT@ECT, Sheri Thomas/HOU/ECT@ECT	27	38	0.11
13	Alan Aronowitz/HOU/ECT@ECT, Roger Balog/HOU/ECT@ECT	20	24	0.07

Secret statistics.

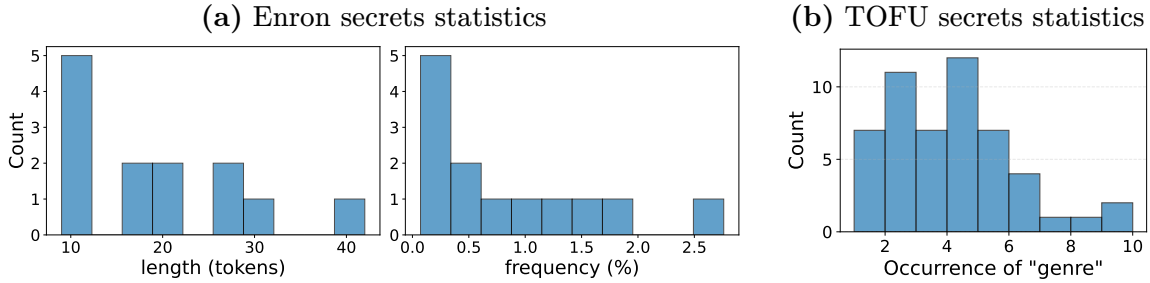


Figure 37: **Secret statistics**: **(a)** Length and frequency of the final set of 13 secrets in Enron. **(b)** Occurrence (frequency) of the secrets among all (20) records per author.

- *Enron.* The secret set size is 13 (full list in Table 10). As shown in Fig. 37, the token lengths of secrets range from 10 to 40, and their frequency (the ratio of the number of samples a secret appears in to the train set size n) varies between 0.07% and 3%.

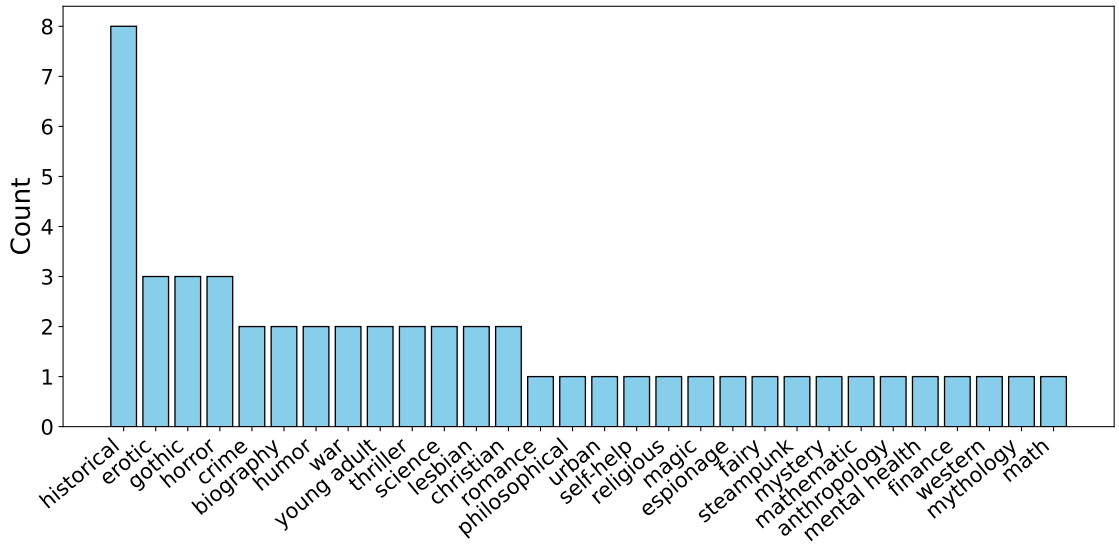


Figure 38: **TOFU secrets**: Histogram of 30 genres across 52 authors in TOFU.

- *TOFU.* The secret set size is 52. Each secret is a mapping from an author to their associated `genre`, which typically consists of one or two words. The frequency of the secrets can be found in Fig. 37. For more than 75% of the authors, their secret appears for fewer than 5 times; the average occurrence is 3.65 times. There are in all 30 unique genres across the 52 authors, as presented in Fig. 38.

Discussion on the privacy unit. Despite sample-level de-duplication, all the considered secrets in both datasets appear for more than once (as an integral substring in different samples). We report the frequency in Fig. 37 and Table 10. This may seem misaligned with the standard sample-level DP we adopt. Nevertheless, the choice of privacy unit matters less in our study, as we focus on variance rather than whether ε provides sufficient privacy protection. Additionally, Enron is not easily partitioned by user—while an email has one sender, it could have multiple receivers, and could be quoting text from emails from a different sender. For the purpose of consistency, we do not use user-level DP [237, 238],

and stick to sample-level DP in our studies.

C.2.7 Empirical Privacy Measures

We describe additional details about the three empirical privacy measures.

ACR. A stochastic search procedure, Greedy Coordinate Gradient [GCG, 239], is adopted to solve the optimization problem of finding the prompt p that makes the model produce a target string s. To account for randomness in this search, we repeat the process with 3 different random seeds. Each run yields a candidate prompt $p^*_{\xi_i}$, which is the shortest prompt found under seed ξ_i (though not guaranteed to be globally optimal). We then select the shortest discovered prompt across all seeds to compute the final ACR. Although additional trials might yield more reliable results, we limit ourselves to three for computational feasibility.

Formally, for a target string s, let $p^*_{\xi_i}$ denote the shortest prompt found by GCG under seed ξ_i. The final ACR score is:

$$\mathrm{ACR}(s) = \max_{i \in \{1,2,3\}} \frac{|s|}{|p^*_{\xi_i}|},$$

where $|\cdot|$ represents the string length.

VMR. The verbatim memorization ratio (VMR) evaluates whether prompting with the prefix s_1 of a secret leads to the recovery of the remainder s_2. We present the prompts and continuations used in measuring VMR in Table 11. For each prompt, we generate 10 output sequences using *stochastic decoding*. We adopt the default parameters in HuggingFace[33] (`temperature`=1.0, `top_p`=1.0, `top_k`=50). We set `max_new_tokens` to match the token length of s_2 for each secret s, ensuring the model has just enough space to generate the expected continuation.

[33] `https://huggingface.co/docs/transformers/main/en/main_classes/text_generation#transformers.GenerationConfig`

Formally, the VMR score for a secret $s = s_1 \| s_2$ is computed as the fraction of generated outputs that exactly match s_2, using a sample size of 10

$$\mathrm{VMR}(s; s_1, s_2) = \frac{1}{10} \sum_{i=1}^{10} \mathbb{1} \left[M(s_1; \texttt{kwargs}, \xi_i) = s_2 \right],$$

where $M(s_1; \texttt{kwargs}, \xi_i)$ denotes the i-th output generated by the model for the prompt s_1, with `kwargs` denoting the decoding parameters introduced above and ξ_i the randomness in generation.

Table 11: **Table of the secrets in Enron.** Each row consists one secret s, as well as the considered prompt s_1 and continuation s_2 for it.

No.	Secret s	Prompt s_1	Continuation s_2
1	Carol St. Clair EB 3889 713-853-3989	Carol St. Clair EB 3889 713	-853-3989
2	Carol St. Clair EB 3892 713-853-3989 (Phone) 713-646-3393 (Fax) carol.st.clair@enron.com	Carol St. Clair EB 3892 713-853-3989 (Phone) 713-646	-3393 (Fax) carol.st.clair@enron.com
3	713-853-5620 (phone) 713-646-3490 (fax) sara.shackleton@enron.com	713-853-5620 (phone) 713-646	-3490 (fax) sara.shackleton@enron.com
4	Richard Shapiro/NA/Enron@Enron, James D Steffes/NA/Enron@Enron	Richard Shapiro/NA/Enron@Enron, James D	Steffes/NA/Enron@Enron
5	Vince J Kaminski/HOU/ECT@ECT, Shirley Crenshaw/HOU/ECT@ECT	Vince J Kaminski/HOU/ECT@ECT, Shirley	Crenshaw/HOU/ECT@ECT
6	Alan Aronowitz/HOU/ECT@ECT, Roger Balog/HOU/ECT@ECT	Alan Aronowitz/HOU/ECT@ECT, Roger	Balog/HOU/ECT@ECT
7	Jones/HOU/ECT@ECT, Samuel Schott/HOU/ECT@ECT, Sheri Thomas/HOU/ECT@ECT	Jones/HOU/ECT@ECT, Samuel Schott/HOU/ECT@ECT, Sheri	Thomas/HOU/ECT@ECT
8	Karen Lambert/HOU/ECT@ECT, Tana Jones/HOU/ECT@ECT	Karen Lambert/HOU/ECT@ECT, Tana	Jones/HOU/ECT@ECT
9	Forwarded by Richard B Sanders/HOU/ECT	Forwarded by Richard B	Sanders/HOU/ECT
10	Forwarded by Steven J Kean/NA/Enron	Forwarded by Steven	J Kean/NA/Enron
11	Forwarded by Mark Taylor/HOU/ECT	Forwarded by Mark	Taylor/HOU/ECT
12	Forwarded by Daren J Farmer/HOU/ECT	Forwarded by Daren J	Farmer/HOU/ECT
13	Forwarded by Jeff Dasovich/SFO/EES	Forwarded by Jeff	Dasovich/SFO/EES

AIR. The AIR metric evaluates whether the ground-truth attribute $A(x)$ is present in the model's output. Specifically, we generate 10 output sequences for each input prompt using *stochastic decoding*, to provide multiple opportunities for the model to reveal the attribute without excessively sampling (which could result in spurious matches). For generation, we

adopt the default parameters in HuggingFace[34] (`temperature`=1.0, `top_p`=1.0, `top_k`=50) and set `max_new_tokens`=20.

Formally, the AIR score for an input x is computed by checking if $A(x)$ appears in at least one of the 10 generated sequences:

$$\text{AIR}(x) = \mathbb{1}\left[\bigvee_{i=1}^{10} A(x) \text{ appears in } M(\mathcal{P}(x); \texttt{kwargs}, \xi_i)\right],$$

where $M(\mathcal{P}(x); \texttt{kwargs}, \xi_i)$ denotes the i-th output generated by the model for the prompt $P(x)$, with `kwargs` denoting the decoding parameters introduced above and ξ_i the randomness in generation.

C.2.8 Utility Measure

For both scenarios (fine-tuning GPT-2 models on Enron and Llama-2 models on TOFU), the utility measure is the cross-entropy loss on the held-out test set. More specifically, the loss is calculated on the full samples for Enron, but only on the "answer" part in TOFU.

In *Enron*, as described in Appendix C.2.1, we ensure that no secrets appear in the held-out test set. Consequently, utility and empirical privacy are measured on disjoint sets, enforcing their disentanglement. In *TOFU*, as detailed in Appendix C.2.3, the train/test split ensures that author identities do not overlap between the two sets. We measure privacy using subsets of authors from the train set only, while utility is evaluated on the test set. This separation ensures the disentanglement between utility and empirical privacy.

C.2.9 More Details of DP Fine-Tuning

DP fine-tuning packages. We follow standard practices for DP fine-tuning of language models. For GPT-2 models, we use the `dp-transformers`[35] package, which natively supports DP fine-tuning of GPT-2 models with a support for LoRA [163]. For Llama-2

[34] `https://huggingface.co/docs/transformers/main/en/main_classes/text_generation#transformers.GenerationConfig`

[35] `https://github.com/microsoft/dp-transformers`

models, we use `dp_finetuning`[36] which natively supports Llama-2 models, along with LoRA compatibility as well. Both packages implement the DP fine-tuning algorithm in Yu et al. [48]. We adopt the PRV privacy accountant [43] for privacy analysis.

Table 12: **Hyperparameter configurations for different scenarios.** The tuple in the rows for GPT-2 models represent (b, T, η, c). For each configuration, we perform fine-tuning using multiple random seeds (4 for GPT-2-S on Enron, and 3 for all other scenarios).

Scenario	Configurations
GPT-2-S, Enron (total=23)	$(8192, 1000, 3\times10^{-3}, 0.5)$ $(8192, 500, 3\times10^{-3}, 0.5)$ $(8192, 250, 3\times10^{-3}, 0.5)$ $(8192, 125, 3\times10^{-3}, 0.5)$ $(4096, 500, 3\times10^{-3}, 0.5)$ $(4096, 250, 3\times10^{-3}, 0.5)$ $(4096, 125, 3\times10^{-3}, 0.5)$ $(2048, 1000, 3\times10^{-3}, 0.5)$ $(2048, 500, 3\times10^{-3}, 0.5)$ $(2048, 250, 3\times10^{-3}, 0.5)$ $(2048, 125, 3\times10^{-3}, 0.5)$ $(1024, 1000, 3\times10^{-3}, 0.5)$ $(1024, 500, 3\times10^{-3}, 0.5)$ $(1024, 250, 3\times10^{-3}, 0.5)$ $(1024, 125, 3\times10^{-3}, 0.5)$ $(8192, 250, 1\times10^{-3}, 0.5)$ $(8192, 250, 1.5\times10^{-3}, 0.5)$ $(8192, 250, 2\times10^{-3}, 0.5)$ $(8192, 250, 4\times10^{-3}, 0.5)$ $(8192, 250, 6\times10^{-3}, 0.5)$ $(4096, 500, 1.5\times10^{-3}, 0.5)$ $(4096, 250, 1.5\times10^{-3}, 0.5)$ $(4096, 250, 6\times10^{-3}, 0.5)$
GPT-2-L, Enron (total=15)	$(8192, 500, 1\times10^{-3}, 0.5)$ $(8192, 250, 1\times10^{-3}, 0.5)$ $(8192, 125, 1\times10^{-3}, 0.5)$ $(4096, 1000, 1\times10^{-3}, 0.5)$ $(4096, 500, 1\times10^{-3}, 0.5)$ $(4096, 250, 1\times10^{-3}, 0.5)$ $(4096, 125, 1\times10^{-3}, 0.5)$ $(2048, 1000, 1\times10^{-3}, 0.5)$ $(2048, 500, 1\times10^{-3}, 0.5)$ $(2048, 250, 1\times10^{-3}, 0.5)$ $(2048, 125, 1\times10^{-3}, 0.5)$ $(1024, 500, 1\times10^{-3}, 0.5)$ $(1024, 125, 1\times10^{-3}, 0.5)$ $(4096, 500, 5\times10^{-4}, 0.5)$ $(4096, 500, 2\times10^{-3}, 0.5)$
Llama-2-7b, TOFU-1 (total=60)	$\{(b, T, \eta, c) \mid b \in \{256, 512, 1024, 2048\}, T \in \{16, 32, 64, 128, 256\}, \eta \in \{5\times10^{-4}, 1\times10^{-3}, 2\times10^{-3}\}, c = 0.5\}$
Llama-2-7b, TOFU-2 (total=60)	$\{(b, T, \eta, c) \mid b \in \{256, 512, 1024, 2048\}, T \in \{16, 32, 64, 128, 256\}, \eta \in \{5\times10^{-4}, 1\times10^{-3}, 2\times10^{-3}\}, c = 0.5\}$
Llama-2-7b, TOFU-4 (total=75)	$\{(b, T, \eta, c) \mid b \in \{256, 512, 1024, 2048, 4096\}, T \in \{16, 32, 64, 128, 256\}, \eta \in \{5\times10^{-4}, 1\times10^{-3}, 2\times10^{-3}\}, c = 0.5\}$

Full set of hyperparameters. Table 12 provides a summary of the hyperparameter configurations used in each scenario. We perform extensive hyperparameter tuning in the space of (b, T, η) while fixing c to a small constant. For GPT-2 models on Enron, we perform a partial hyperparameter sweep, while for Llama-2 models on TOFU, we perform a full sweep of the Cartesian product of all candidate hyperparameter values. Below, we explain the rationale for this distinction.

Fine-Tuning on Enron vs. TOFU. Fine-tuning on Enron requires significantly larger compute C (as can be seen in the table). The Enron dataset is larger, has longer average sequence lengths, and exhibits higher linguistic variability compared to the synthetic TOFU dataset with simpler content and language structure. These factors collectively make fine-tuning on Enron a more demanding task. Additionally, we adopt different training precisions based on the native setups of the fine-tuning frameworks: *fp32* for GPT-2 models

[36] `https://github.com/google-research/google-research/tree/master/dp_instructions/dp_finetuning`

using `dp-transformers` and *bfloat16* for Llama-2 models using `dp_finetuning`.

Partial hyperparameters sweep for GPT-2 models. For GPT-2 models on Enron, we first conduct a coarse-grained grid search to identify a strong candidate configuration $(b^\star, T^\star, \eta^\star)$, which is $(8192, 250, 3 \times 10^{-3})$ for GPT-2-S and $(4096, 500, 10^{-3})$ for GPT-2-L. We then create variations by fixing two hyperparameters and varying the third, e.g., $(b^\star, T^\star/2, \eta^\star)$, $(2b^\star, T^\star, \eta^\star)$. We further include other configurations to ensure decent coverage of the hyperparameter space.

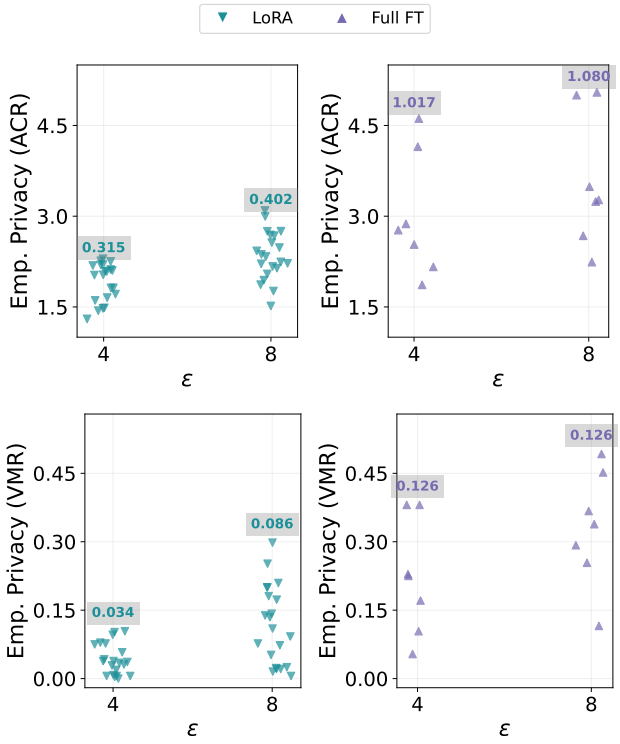


Figure 39: **Empirical privacy variance across different fine-tuning paradigms.** *Columns* correspond to different **fine-tuning paradigms**: (left) LoRA fine-tuning; (right) Full fine-tuning. Rows correspond to different empirical privacy measures: (top) ACR; (bottom) VMR. Each subfigure shows empirical privacy scores achieved by models trained using different configuration at different ε's. Each group's standard deviation is labeled at the top of its cluster. The results show that full fine-tuning exhibits higher empirical privacy variance than LoRA fine-tuning for both measures (comparing the two columns).

C.3 Additional Experimental Results for Sec. 5.2

C.3.1 Additional Results on empirical privacy variance

Additional results on trends.

Fine-tuning paradigm (Fig. 39). We compare LoRA fine-tuning [163] and full fine-tuning for GPT-2-S at $\varepsilon = 4$ and $\varepsilon = 8$, evaluating their ACR and VMR. The results show that full fine-tuning has higher empirical privacy variance than LoRA fine-tuning for both measures. We note that the variance increase from $\varepsilon = 4$ to $\varepsilon = 8$ is less pronounced in full fine-tuning.

We conjecture that this is due to the limited number of configurations explored in our full fine-tuning experiments.

Model size (Fig. 40). We compare Llama-2-7b and Llama-2-13b on TOFU-2 at $\varepsilon = 8$, evaluating their AIR. The results indicate that larger models have higher empirical privacy variance, consistent with the findings in Sec. 5.2.2.

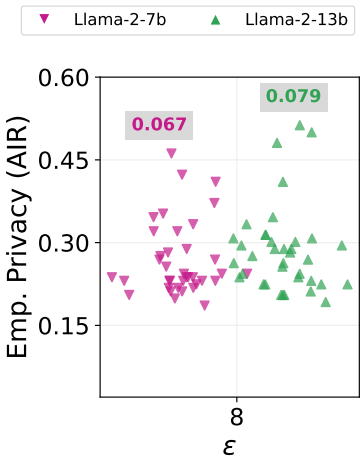


Figure 40: **Empirical privacy variance under different model sizes (Llama-2-7b vs. Llama-2-13b).** Each color corresponds to a model size. The figure shows AIR scores achieved by models trained using different configurations. Each group's standard deviation is labeled at the top of its cluster. The results show that Llama-2-13b achieves higher empirical privacy variance than Llama-2-7b.

Scaling dataset size while maintaining secret count (Fig. 41). In Sec. 5.2, we study scaling the dataset size while maintaining the secret *density*. Fig. 14(b) shows that increasing dataset size in this way leads to increased empirical privacy. As a complementary study, we investigate scaling the dataset size while maintaining the secret *count*. Concretely, we generate another 200 synthetic author profiles and merge them with TOFU-1. We refer to the obtained dataset as TOFU-2*. Compared to TOFU, TOFU-2* has doubled dataset size but the same secret count; compared to TOFU-2, TOFU-2* has the same dataset size but half secret density. We present the comparison between TOFU, TOFU-2 and TOFU-2* in Fig. 41. TOFU-2* achieves the lowest empirical privacy among all.

Additional results on consistency of the trends.

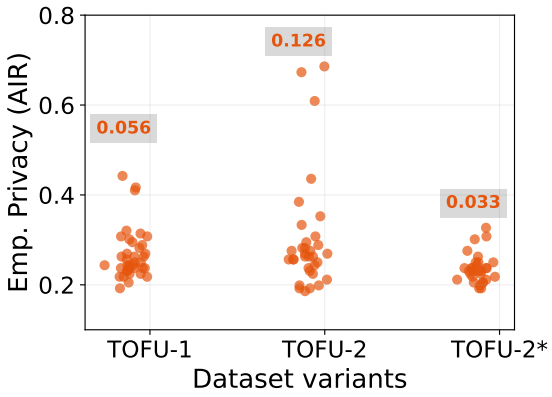


Figure 41: **Empirical privacy variance under different data variants (TOFU, TOFU-2, and TOFU-2*)**, at $\varepsilon = 8$. The figure shows AIR scores achieved by models trained using different configurations. Each group's standard deviation is labeled at the top of its cluster. TOFU-2* achieves the lowest empirical privacy variance among the three.

Secret subset (Fig. 42). We conduct this experiment using Llama-2-7b and variants of the TOFU dataset. We randomly sample half of the secrets (26 out of 52 author-genre pairs) without replacement for three times to create subsets (0, 1, 2). We then measure AIR of these subsets for models trained on different dataset sizes (TOFU-1, TOFU-2, TOFU-4) with varying ε's. The results show that, across all subsets considered, empirical privacy variance increases as either ε or dataset size grows.

Empirical privacy measure (Fig. 43). We conduct this experiment on the Enron dataset. We measure ACR and VMR for models of different sizes (GPT-2-S and GPT-2-L) trained with varying ε's. The results show that, for both empirical privacy measures, empirical privacy variance increases as either ε or model size grows.

C.3.2 Does Model Distance Explain the Trends of empirical privacy?

One plausible hypothesis for the observed trends in empirical privacy variance is that, a set of models exhibits high empirical privacy variance because the average "distance" (which we will formally define shortly) between them is also large. The intuition behind this hypothesis is fairly straightforward: models that are "close" to each other should have similar empirical privacy scores. This hypothesis can be formally stated as follows: if the

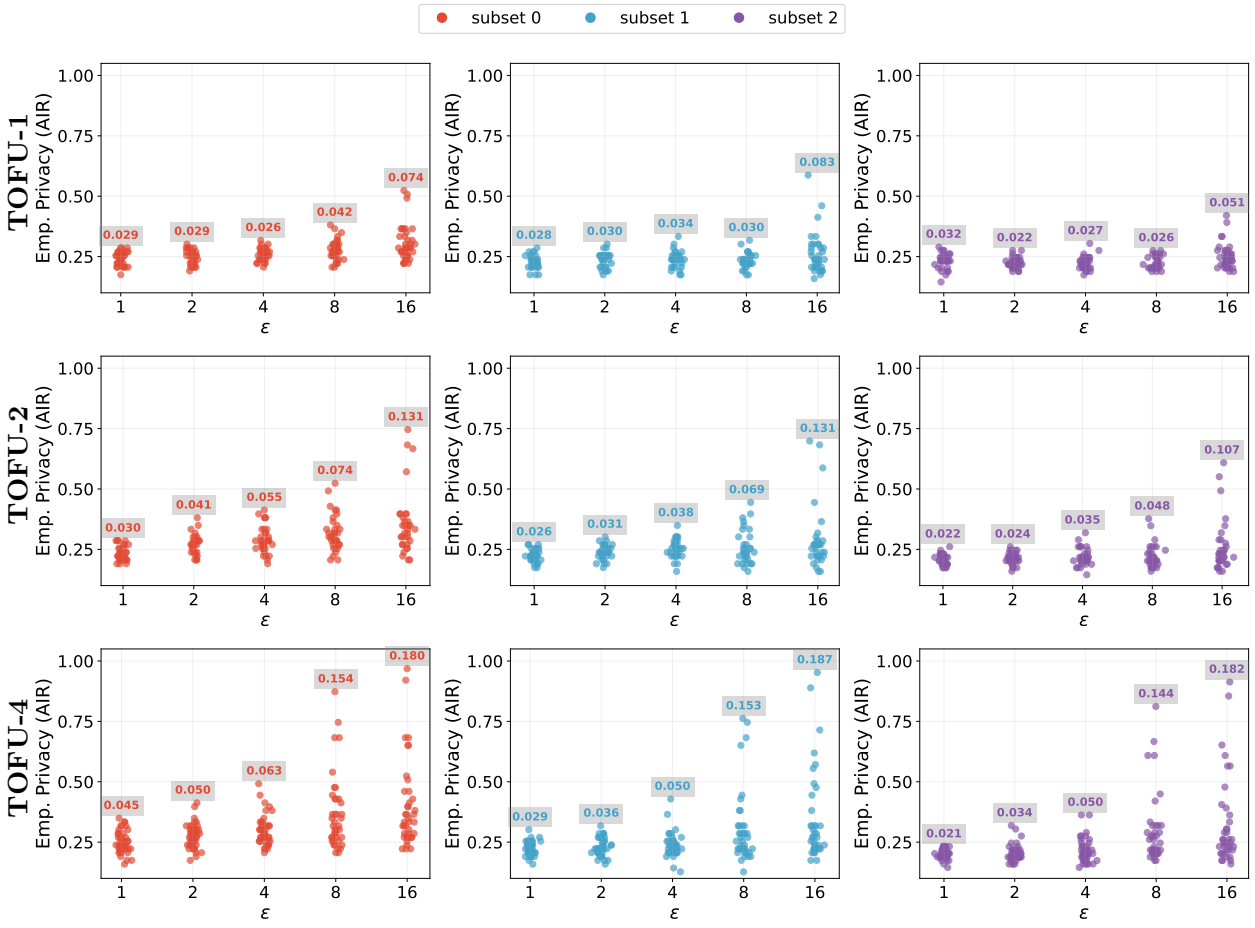


Figure 42: **Empirical privacy variance across different secret subsets.** *Columns* correspond to different subsets, and rows correspond to different dataset sizes (TOFU-1, TOFU-2, TOFU-4 from top to bottom). Each subfigure shows AIR values achieved by models trained using different configuration at different ε's. Each group's standard deviation is labeled at the top of its cluster. The results show that, across all subsets considered, empirical privacy variance increases as either ε or dataset size grows.

average "distance" within S_1 is greater than that within S_2, then the empirical privacy variance measured on S_1 will also be larger than that on S_2. Here, S_1 and S_2 denote two sets of models, where models within each set share the same architecture and initialization, are trained on the same dataset using DP-SGD, and differ only in their configurations and inherent training randomness.

Choices of (S_1, S_2) pairs. We consider three types of (S_1, S_2) pairs corresponding to the three types of trends we identify in Sec. 5.2.2: 1) S_1 and S_2 share model and data, but

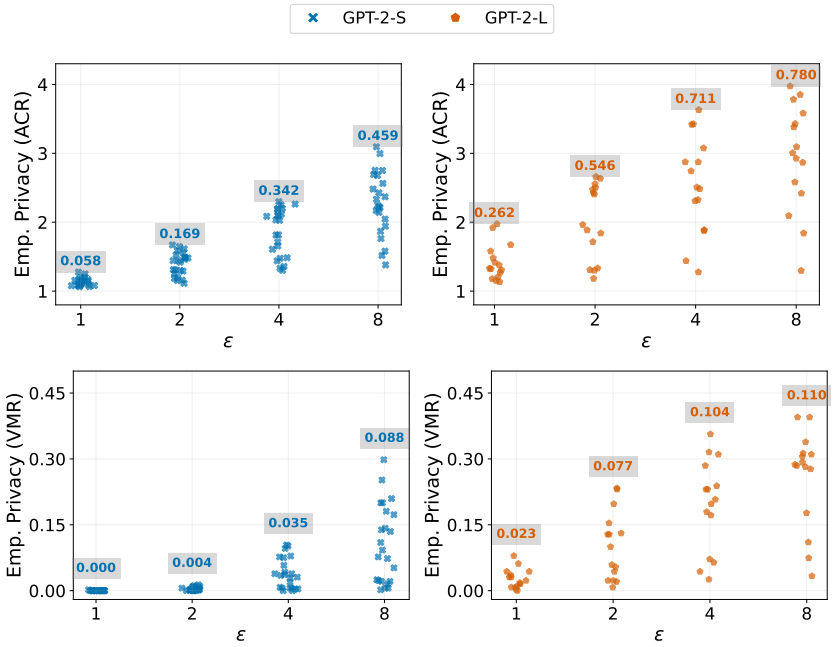


Figure 43: **Empirical privacy variance across different empirical privacy measures (ACR and VMR).** *Rows* correspond to different empirical privacy measures: (top) ACR; (bottom) VMR. Columns correspond to different model sizes: (left) GPT-2-S; (right) GPT-2-L. Each subfigure shows empirical privacy scores achieved by models trained using different configuration at different ε's. Each group's standard deviation is labeled at the top of its cluster. The results show that, for both empirical privacy measures, empirical privacy variance increases as either ε or model size grows.

differ in ε; 2) S_1 and S_2 share ε and data, but differ in model size; 3) S_1 and S_2 share ε and model, but differ in dataset size.

Distance metrics. We compute the average distance over a set of models by the *mean pairwise* distance over all *model pairs*. We consider two distance metrics: 1) parameter space distance—ℓ_2 distance between model parameters; 2) functional distance—ℓ_2 distance between model's prediction on a held-out test set, formally:

$$d_f(M_1, M_2) \;=\; \mathbb{E}_{x\sim\mathcal{D},t\sim[T]}\Big[\big\|\mathrm{softmax}\big(M_1(x)_t\big) \;-\; \mathrm{softmax}\big(M_2(x)_t\big)\big\|_2\Big], \tag{19}$$

where M_1 and M_2 are two models, D stands for the empirical distribution of the held-out test set, T is the number of token positions, and $\mathrm{softmax}(M(x)_t)$ denotes the probability

distribution over the vocabulary obtained by applying softmax to the logits.

Results. Table 13 shows that: 1) holding data and model fixed, *larger* ε has *smaller* average distance; 2) holding data and ε fixed, *larger* model size has *smaller* average distance; 3) holding model and ε fixed, varying dataset size does not seem to affect the average distance. These results **refute** the hypothesis and suggest that model distance might not be able to explain the trends of empirical privacy in Sec. 5.2.

Table 13: **Average distance within model sets with varying ε, model size, or dataset size.** We report functional distance in (b-c).

(a) Vary ε (model=GPT-2-S, data=Enron)

ε	Param. Dist.	Func. Dist.
1	32.48	0.1244
2	32.07	0.1143
4	31.74	0.1086
8	31.24	0.1023

(b) Vary model size (data=Enron)

ε	GPT-2-S	GPT-2-L
1	0.1244	0.0920
2	0.1143	0.0863
4	0.1086	0.0824
8	0.1023	0.0796

(c) Vary dataset size (model=Llama-2-7b)

ε	TOFU-1	TOFU-2	TOFU-4
1	0.0580	0.0539	0.0575
8	0.0637	0.0592	0.0632

C.3.3 Impact of Hyperparameters vs. Impact of Random Seeds

For all results reported in Sec. 5.2.2 and in Appendix C.3.1, we averaged out the effect of random seeds. Here, we compare the variance induced by random seeds and that induced by hyperparameter configurations. For seeds, we compute the standard deviation of empirical privacy scores across seeds for each configuration and average these values. For configurations, we compute the mean across seeds for each configuration and then the standard deviation of these means. We present the results in Table 14, showing that variance from seeds is approximately *half* that of configurations.

Table 14: **Variance induced by inherent randomness in model training vs. variance induced by hyperparameters.** The numbers reported in the table are standard deviations.

	GPT-2-S, Enron, ACR		GPT-2-L, Enron, ACR	
ε	randomness	hyperparameter	randomness	hyperparameter
1	0.06	0.06	0.14	0.26
2	0.11	0.16	0.26	0.48
4	0.16	0.32	0.28	0.56
8	0.19	0.41	0.31	0.58

C.4 Additional Experimental Results for Sec. 5.3

C.4.1 Additional Results on Accuracy of Heuristics

Fig. 44 shows the complete set of results corresponding to all combinations of models, datasets, and empirical privacy measures, evaluated at varying ε values.

The results show that our heuristics outperform the random guess baseline. Notably, on the TOFU datasets, heuristic accuracy improves with increasing ε and larger dataset sizes (see Fig. 44(3a–3e)).

For the specific setting of (GPT-2-S, Enron, VMR, $\varepsilon = 1$), the accuracy of all three heuristics is close to 0. This is due to an artifact where the VMR scores are nearly all 0 for all configurations at $\varepsilon = 1$, meaning no configuration is distinguishably better than another, leading to the observed low accuracy.

More evaluation results on different density groups of *density-adjusted* TOFU can be found in Fig. 53 in Appendix C.6.

C.4.2 Visualization of Selection Quality

Fig. 45(a) illustrates the layout of empirical privacy and utility for all configurations, serving as the basis of our selection process. We progressively slide a utility threshold u from left to right (high to low utility), and at each threshold, each selection method chooses a configuration from the corresponding subpool P_u.

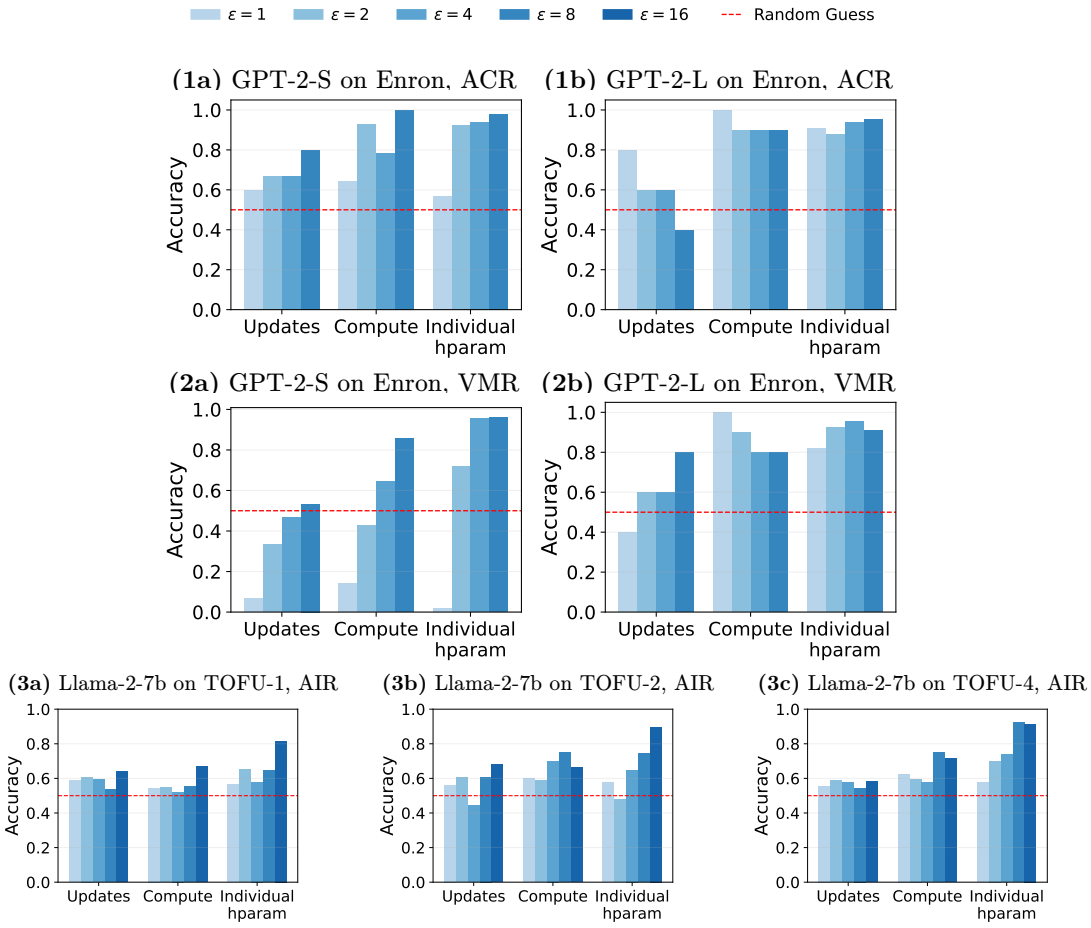


Figure 44: Accuracy of the three heuristics across different **models**, **datasets**, and **empirical privacy measures**, evaluated at varying ε's.

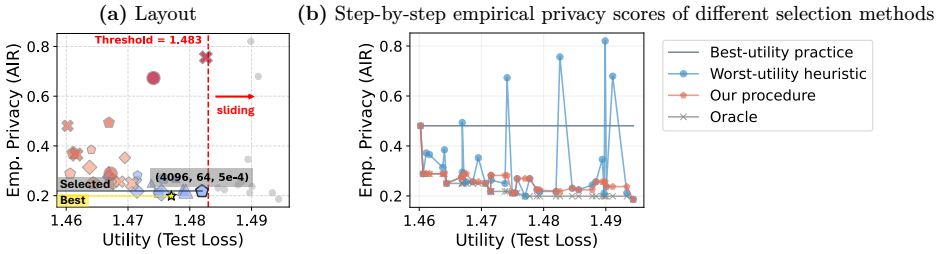


Figure 45: Setting=(Llama-2-7b, TOFU-4, $\varepsilon = 8$). **(a)** The layout of empirical privacy (AIR) vs. utility (test loss) achieved by all configurations calibrated to the same DP guarantee. **(b)** At each utility threshold u, each of the considered method (oracle, best-utility practice, worst-utility heuristic, and our procedure) will make a selection from the subpool P_u. We plot the scores of the selected points against the thresholds.

Results. Fig. 45(b) presents the empirical privacy scores (AIR) of configurations selected by different methods across all thresholds. The visualization offers a more fine-grained and intuitive comparison of the selection quality of different methods. As a reference, the oracle points collectively form the Pareto front. We observe that the **best-utility practice** prioritizes utility at the cost of empirical privacy, while the **worst-utility heuristic** appears unstable and overly sensitive to individual points. In contrast, **our procedure** exhibits near-oracle behavior, ensuring stable and robust performance across all threshold levels. A full set of demonstration results can be found at Figs. 50 to 52 in Appendix C.6.

C.4.3 Additional Results on Relative Privacy Risk

We evaluate all combinations of **models**, **datasets**, **empirical privacy measures**, and ε **values**. Each combination corresponds to a specific layout of models for the hyperparameter selection task (see Fig. 45(a)). Before discussing the results, we introduce additional considerations in the experimental setup.

Accounting for training randomness. In Fig. 45(a), each point represents an average over multiple random seeds, meaning the observed layout is just one realization drawn from an underlying distribution. This simplification averages out the impact of training randomness.

To better account for the variation in privacy risks, we sample layouts using a Monte Carlo approach. Specifically, we model each configuration as a Gaussian distribution, with its mean and standard deviation estimated from empirical data (i.e., across random seeds). This allows us to generate multiple plausible layouts and analyze the effectiveness and robustness of our selection method under training randomness. In our experiments, we adopt the number of trials as 5,000.

A complementary metric: absolute privacy risk The relative privacy risk metric introduced in the main paper runs into issues when the oracle's privacy risk is zero, which occurs in the case of VMR. Since relative comparisons become ill-defined in such cases, we also compute an absolute privacy risk measure to ensure meaningful evaluations across all settings.

Results. The results are presented in Figs. 46 to 48. Our procedure outperforms the baselines in almost all settings, demonstrating the effectiveness of the underlying heuristics and their ability to generalize. More evaluation results on different density groups of *density-adjusted* TOFU can be found in Fig. 54 in Appendix C.6.

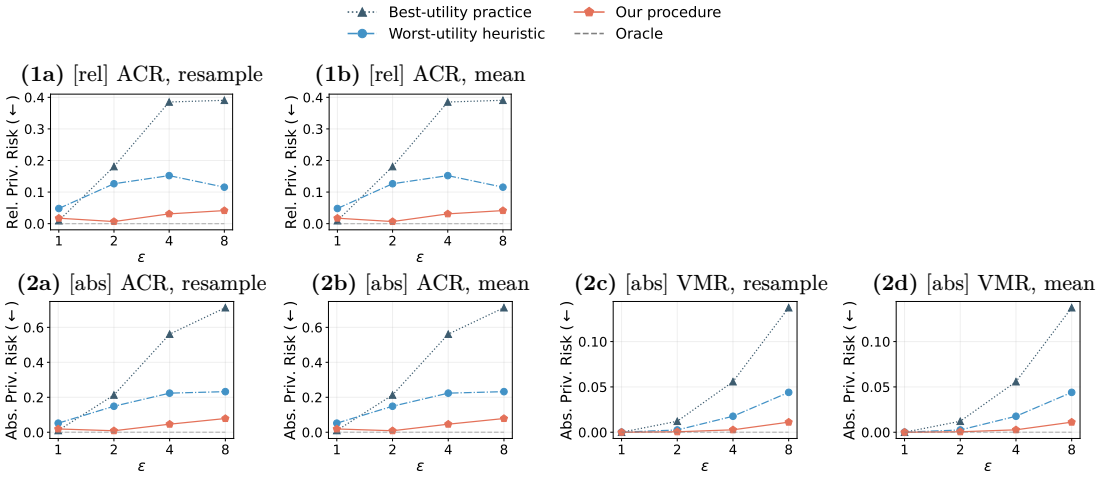


Figure 46: **Relative privacy risk of our procedure compared with baselines.** Model = GPT-2-S; data = Enron; empirical privacy measure ∈ {ACR, VMR}; risk measure ∈ {abs, rel}, where "abs" denotes the absolute privacy risk and "rel" denotes for the relative privacy risk; layout ∈ {resample, mean}, where "resample" corresponds to the monte carlo approach that accounts for the training randomness, and "mean" corresponds to the approach that directly uses the mean, averaging out the training randomness. *Note that GPT-2-S achieves VMR of 0 at multiple configurations, which makes the relative risk metric invalid. Thus we omit the relative risk results for VMR and present results for the absolute risk only.*

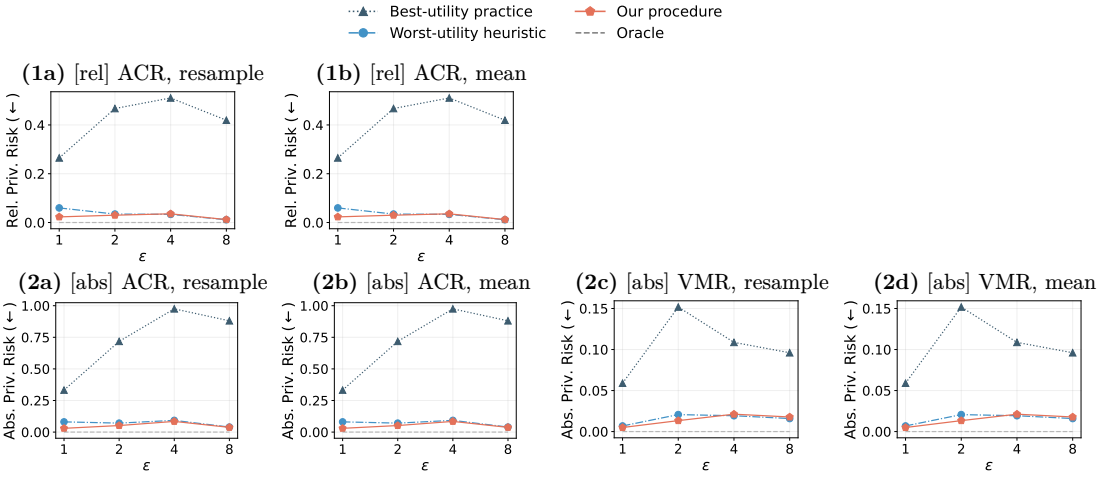


Figure 47: **Relative privacy risk of our procedure compared with baselines.** Model = GPT-2-L, data = Enron, empirical privacy measure ∈ {ACR, VMR}; risk measure ∈ {abs, rel}; layout ∈ {resample, mean}. *Note that GPT-2-L achieves VMR of 0 at multiple configurations, which makes the relative risk metric invalid. Thus we omit the relative risk results for VMR and present results for the absolute risk only.*

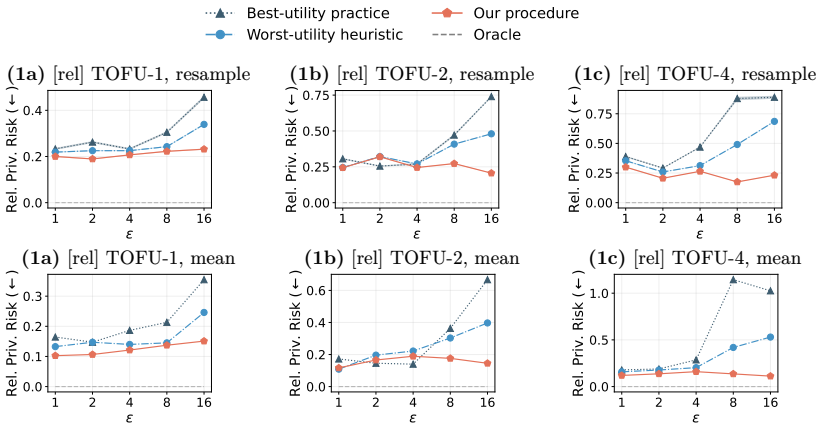


Figure 48: **Relative privacy risk of our procedure compared with baselines.** Model = Llama-2-7b, data ∈ {TOFU-1, TOFU-2, TOFU-4}, measure = AIR, risk = rel, layout = resample.

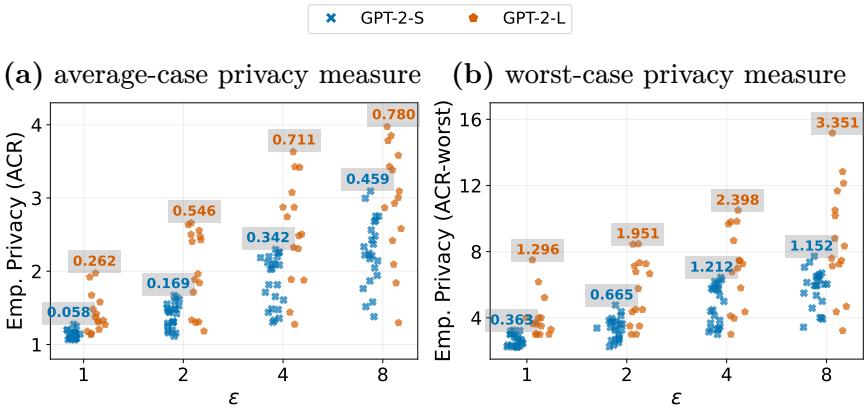


Figure 49: **Empirical privacy variance happens for both average-case (left) and worst-case (right) privacy measures.**

C.5 Results under the Worst-Case Privacy Measure

In this section, we investigate whether switching to a worst-case empirical privacy measure has any impact on the conclusions we obtained in the main paper. We achieve this by taking the max of the measured empirical privacy scores on the set of secrets, instead of the average as done in the main paper (see the end of Sec. 5.2.1).

We present detailed results below, supporting that all conclusions remain unchanged.

Empirical privacy. Fig. 49 shows that empirical privacy variance happens for both the average-case and the worst-case privacy measure (the former is the same as Fig. 14(a)).

Regression results. In Table 15, we present the regression results using the *worst-case* privacy measure as the response variable y, and compare with that obtained using the *average-case* privacy measure (as presented in Table 2 in the main paper).

As shown in Table 15, the conclusions drawn on the *average-case* regression results (in Sec. 5.3.1) remain to hold on the *worst-case* regression results—(1) all individual hyperparameters have positive coefficients with significant p-values, thus increasing any individual hyperparameter leads to worse empirical privacy; (2) The coefficient of $\log b$ is smallest, meaning that under fixed compute, increasing b (while decreasing T proportionally) improves empirical privacy; (3) The coefficient of $\log \eta$ is larger than $\log C$, meaning under fixed updates, decreasing η (while increasing C proportionally) improves empirical privacy.

Table 15: **Comparison on regression results on the average-case privacy measure vs. the worst-case privacy measure.**

(a) Regression on *individual* hyperparameters

	Enron *(average-case)*		Enron *(worst-case)*	
Variable	Coef.	p-value	Coef.	p-value
Batch size ($\log b$)	0.13^{***}	1×10^{-5}	0.38^{**}	2×10^{-3}
Iterations ($\log T$)	0.37^{***}	$< 2 \times 10^{-16}$	1.31^{***}	5×10^{-14}
Learning rate ($\log \eta$)	0.51^{***}	5×10^{-15}	1.80^{***}	9×10^{-12}

(b) Regression on *composite* hyperparameters

	Enron *(average-case)*		Enron *(worst-case)*	
Variable	Coef.	p-value	Coef.	p-value
Compute ($\log C$)	0.22^{***}	2×10^{-12}	0.74^{***}	3×10^{-9}
Learning rate ($\log \eta$)	0.53^{***}	6×10^{-13}	1.89^{***}	2×10^{-10}

Notes: $^{***}p < 0.001$, $^{**}p < 0.01$, $^{*}p < 0.05$. The response variable (empirical privacy score y) is ACR.

Correlation with audited $\hat{\varepsilon}$. We measure the spearman rank correlation between the audited $\hat{\varepsilon}$ and the worst-case empirical privacy score. Similar to the conclusion obtained on the average-case privacy measure, here we obtain a correlation of -0.29 between the two, showing that the two are not positively correlated. Moreover, the correlation between the average-case privacy measure and the worst-case privacy measure is as high as 0.90.

C.6 Complete Sets of Results

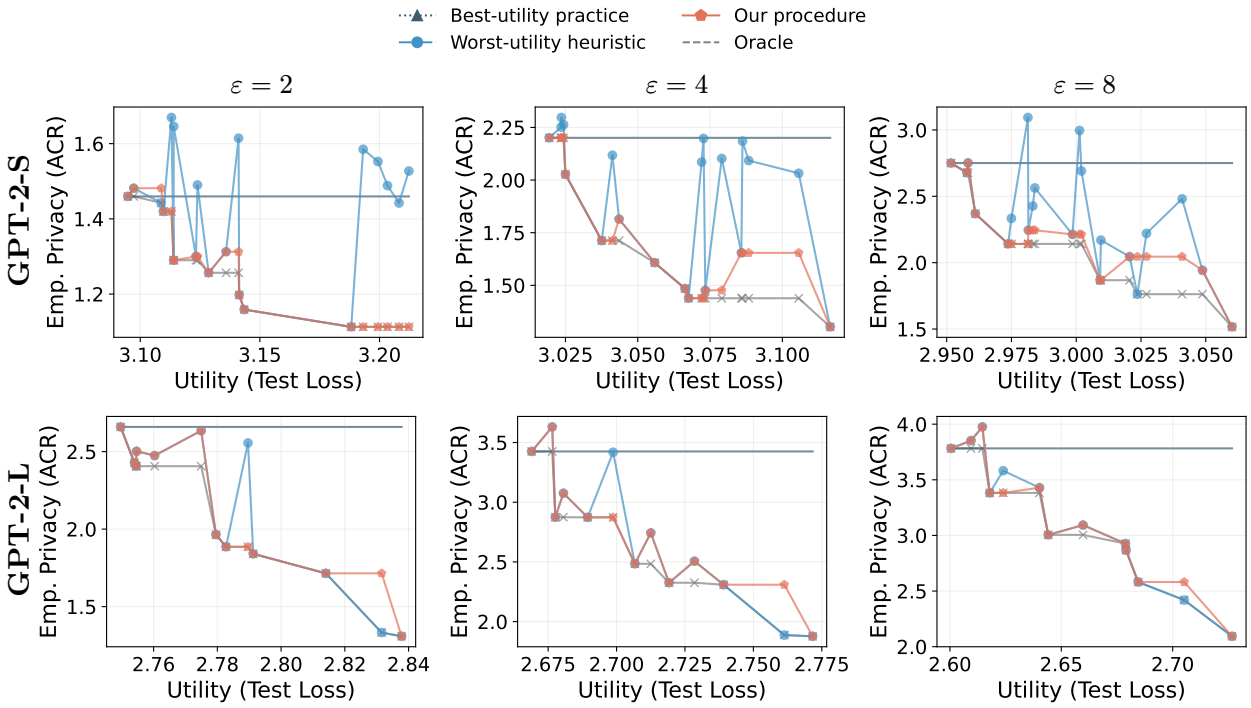


Figure 50: **A complete set of visualizations on the selection quality of different methods. Dataset: Enron; Model: GPT-2 models.** our procedure *consistently outperforms* the others (i.e., best-utility practice and worst-utility heuristic) for both models (GPT-2-S and GPT-2-L) and across different ε values.

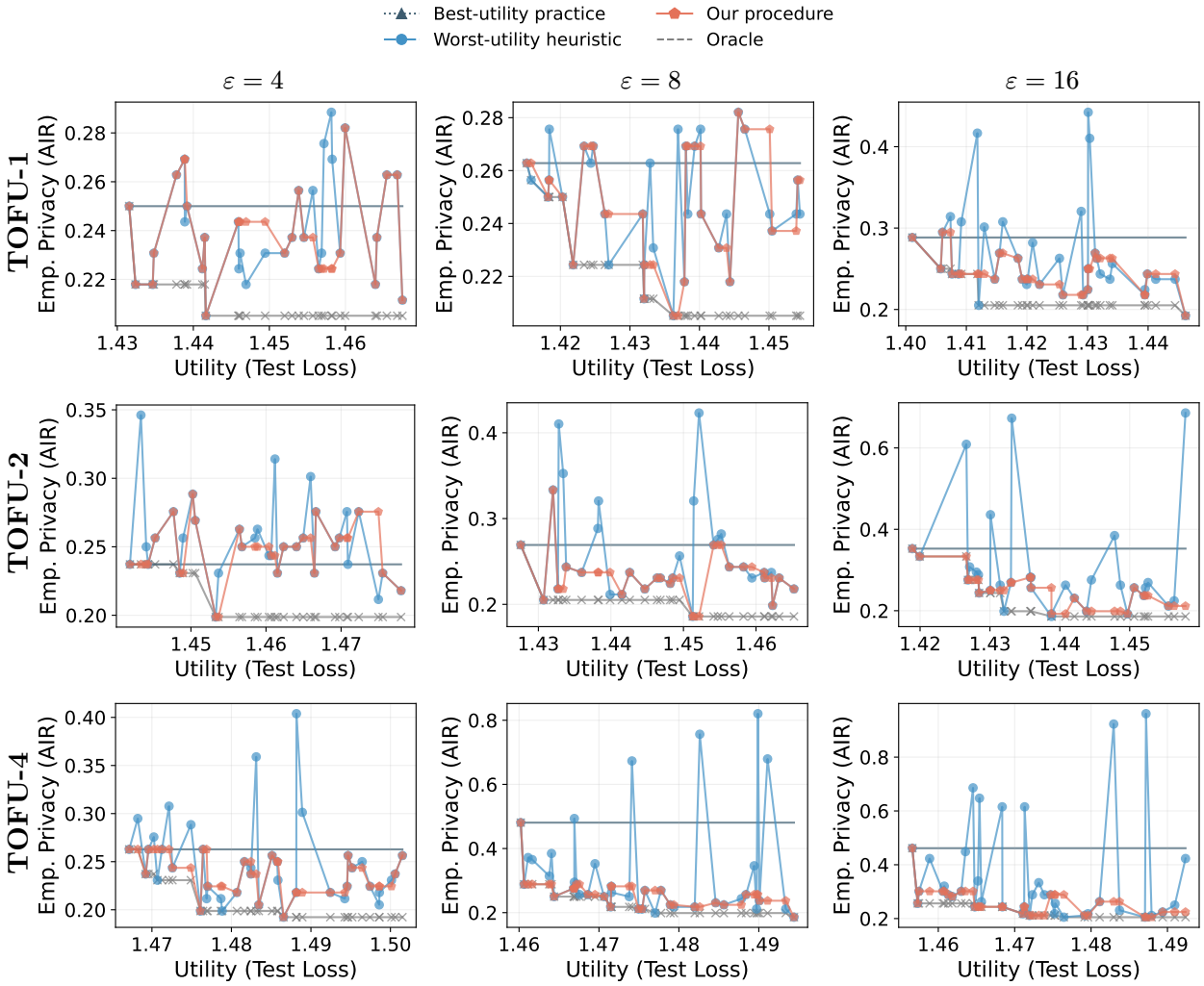


Figure 51: **A complete set of visualizations on the selection quality of different methods. Dataset: *paraphrase-scaled* TOFU; Model: Llama-2-7b.** The advantage of our procedure over the others (i.e., best-utility practice and worst-utility heuristic) enlarges with the increase of the dataset size (vertically, from TOFU-1 to TOFU-4) and the increase of ε (horizontally, from $\varepsilon = 4$ to $\varepsilon = 8$).

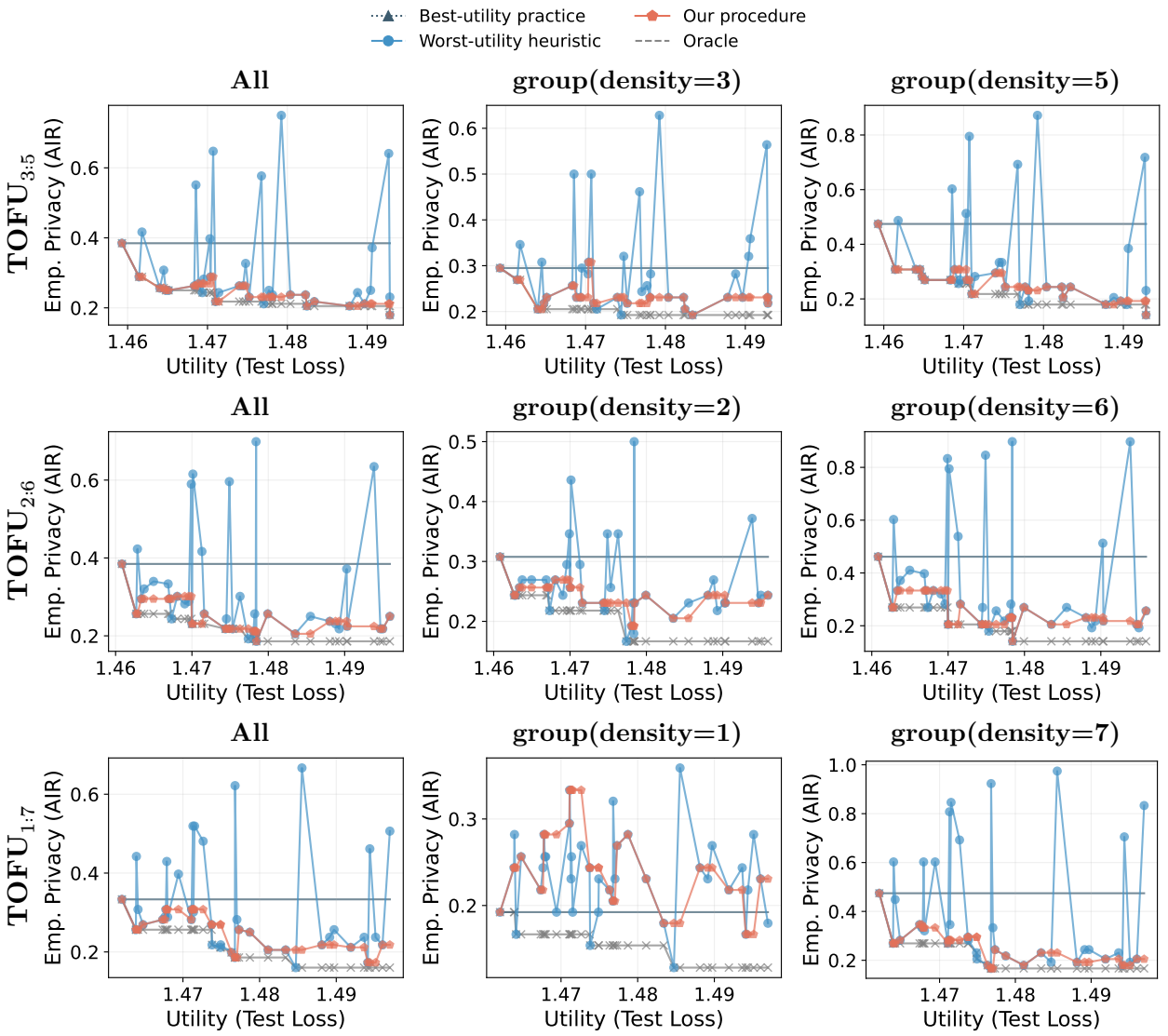


Figure 52: **A complete set of visualizations on the selection quality of different methods. Dataset: *density-adjusted* TOFU**: {$\text{TOFU}_{1:7}$, $\text{TOFU}_{2:6}$, $\text{TOFU}_{3:5}$} (reflected in rows); Model: Llama-2-7b; $\varepsilon = 8$. Groups: all, low- and high-density groups (reflected in columns). Specifically, "density=1" means no augmentation; "density=x" means augmenting the dataset with additional $x-1$ paraphrased texts besides the original one, for each sample. The advantage of **our procedure** over the others (i.e., **best-utility practice** and **worst-utility heuristic**) enlarges with the increase of the secret density. Notably, even at a small density ("density=2", meaning the secret occurs for less than 0.06% in all samples), **our procedure** already starts to demonstrate advantages.

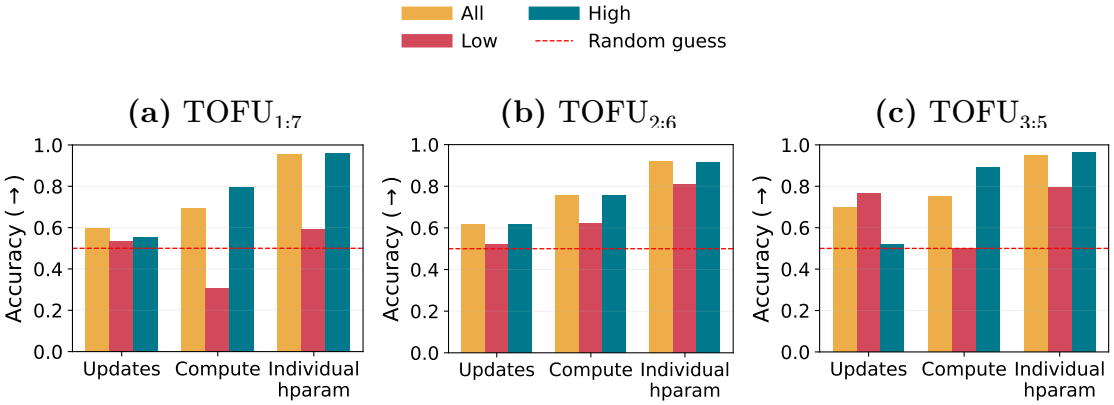


Figure 53: **Accuracy of the three heuristics. Dataset: *density-adjusted* TOFU**: {$\text{TOFU}_{1:7}$, $\text{TOFU}_{2:6}$, $\text{TOFU}_{3:5}$} (reflected in columns); Model: Llama-2-7b; $\varepsilon = 8$; groups = all, low, high density groups; measure = AIR.

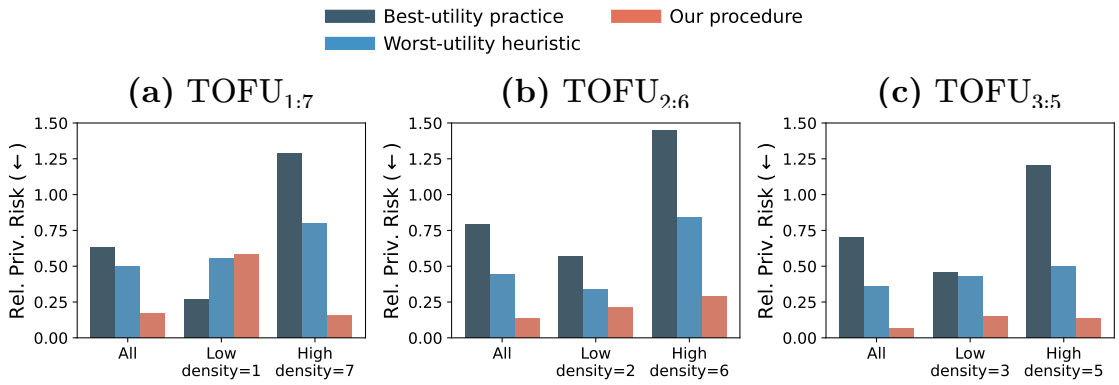


Figure 54: **Relative privacy risk of our procedure compared with baselines. Dataset: *density-adjusted* TOFU**: {$\text{TOFU}_{1:7}$, $\text{TOFU}_{2:6}$, $\text{TOFU}_{3:5}$} (reflected in columns); Model: Llama-2-7b; $\varepsilon = 8$, measure = AIR, risk = rel, layout = mean. (Refer to Fig. 46 for explanations on "rel" and "mean" and Fig. 52 for explanations on "density".) Similarly, we observe that our procedure has a large advantage over the others, which becomes apparent even at low densities (e.g., density=2) and grows as the density increases.

D Appendix for Chapter 6: ACTG-ARL

D.1 Additional Details of Our Approach

D.1.1 LLM-Assisted Schema Design and Extraction

Schema design via LLM. Fig. 55 provides the initial prompt for schema design. Using the template prompt, we fill in the corresponding "dataset_description", "workload_description", and "num_features" for each private dataset domain we want to build a schema for. We optionally supply examples if public or donated examples are available.

Discussion on feature extraction. For $\mathcal{S}_1$, we rely on a pretrained topic model for feature extraction. Since this model can be downloaded and run locally, it poses no risk of privacy leakage. For $\mathcal{S}_2$ and $\mathcal{S}_3$, we instead use a powerful LLM, M_{oracle} (specifically `gemini-2.5-flash-lite`), to perform feature extraction.

In our main experiments, we assume a threat model in which the server-hosted model is trustworthy. This means that sharing data with the server does not lead to a privacy breach, consistent with the policies of major LLM providers[37]. Nevertheless, even if the server behaves adversarially [240], our algorithm remains applicable. In such cases, one option is to deploy an open-source LLM locally for the same task. Alternatively, privacy-preserving inference methods can be adopted to enable secure API queries [241, 242, 243, 244].

Schemas for the two datasets. We derive schemas for the two private datasets in our study following the above and provide their corresponding schema below: Fig. 56 for bioRxiv and Fig. 57 for PMC-patients.

Feature extraction via LLM. After obtaining the schema, we prompt M_{oracle} to extract the features according to the pre-specified schema. Fig. 58 provides the prompt for feature

[37] `https://ai.google.dev/gemini-api/terms#data-use-paid`

extraction.

D.1.2 Privacy Accounting

For each method and each total budget ε, we independently tune the budget split $(\varepsilon_1, \varepsilon_2)$. We use state-of-the-art privacy accountants [43, 44, 245] to calibrate to the final (ε, δ)-DP guarantee. We use different methods for accounting depending on the framework instantiation.

- **CTCL ($\mathcal{S}_1$)**: composition of 1) a Gaussian mechanism (for the DP histogram) with 2) composition of subsampled Gaussian mechanism (for DP-FT). This can be handled with Privacy Loss Distribution (PLD) accountants [43, 44].
- **Conditional generation with free-form feature ($\mathcal{S}_2$)**: composition of 1) composition of subsampled Gaussian mechanism (for DP-FT) and 2) composition of subsampled Gaussian mechanism (for DP-FT). This can be handled with Privacy Loss Distribution (PLD) accountants [43, 44].
- **ACTG ($\mathcal{S}_3$)**: AIM satisfies ρ-zCDP. From the perspective of the RDP accountant, this is interchangeable with a Gaussian machanism (as they have the same RDP guarantees for all α). Thus we treat this as a composition of 1) a Gaussian mechanism (for AIM) and 2) composition of subsampled Gaussian mechanism (for DP-FT), and use RDP accountant [245, 246] to perform accounting.

D.1.3 Details of the RL Algorithm PPO

Proximal Policy Optimization (PPO). PPO [206] is a policy gradient algorithm that maximizes a clipped surrogate objective to stabilize policy updates. Let $\pi_\theta(a \mid s)$ be the policy parameterized by θ, and let $A^\pi(s, a)$ be an estimator of the advantage function. Define the probability ratio

$$r_\theta(s, a) = \frac{\pi_\theta(a \mid s)}{\pi_{\theta_{\text{old}}}(a \mid s)}.$$

The PPO objective is

$$\mathcal{L}^{\text{PPO}}(\theta) = \mathbb{E}_{(s,a)\sim\mathcal{B}}\Big[\min\big(r_\theta(s,a)\,A^\pi(s,a),\ \text{clip}(r_\theta(s,a),\,1-\epsilon,\,1+\epsilon)\,A^\pi(s,a)\big)\Big],$$

where $\mathcal{B}$ is a batch of transitions collected using the old policy $\pi_{\theta_{\text{old}}}$, and $\epsilon > 0$ controls the amount of clipping to limit policy changes and promote stable learning.

PPO in language models. In LLM training, PPO is commonly implemented through the `TRL` (Transformer Reinforcement Learning) framework [247], which adapts the standard PPO update to sequence models by operating on token-level log-probabilities and performing rollouts in text space. In our implementation, we follow the TRL PPO pipeline but *replace the default LM-based reward model with our rubric reward* as detailed in Sec. 6.3.1. We further adapt the pipeline to optimize a hybrid objective of our Anchored RL (Sec. 6.3.2). We provide implementation details and experimental setups of ACTG-RL and ACTG-ARL in Appendix D.3.4.

D.2 Full Algorithm and Pseudocode

We provide the pseudocode of the full algorithm in Alg. 6, which includes the following stages.

1. **Private data annotation**: First, we annotate the private dataset with a structured tabular schema ($\mathcal{S}_3$) via inference calls to M_{oracle}.

2. **Initial DP generators training:** We then train the initial DP generators: the feature generator (G_f) using AIM, and the conditional text generator ($G_{x|f}$) using DP-FT.

3. **Anchor dataset curation:** Using the initial generators, we curate a high-quality synthetic dataset D_{SFT_N} via best-of-N sampling.

Algorithm 6: ACTG-ARL

Input : Private dataset D^x_{priv}, LLM feature extractor M_{oracle}, `AIM` parameter ρ, `DP-FT` parameter σ, best-of-N parameter N

Output : Feature generator G_f, final conditional text generator $G^{\text{ARL}}_{x|f}$, generated synthetic features $D^{\tilde{f}}_{\text{syn}}$, generated synthetic text $D^{\tilde{x}}_{\text{syn}}$

// Step 1: Annotate private data

$D^f_{\text{priv}} \leftarrow \texttt{Annotate}(M_{\text{oracle}}, D^x_{\text{priv}})$ ▷ Extract features according to a rich schema

// Step 2: Train initial DP feature generator and conditional generator

$G_f \leftarrow \texttt{AIM}(D^f_{\text{priv}}, \rho)$ ▷ Obtain a DP feature generator for synthetic features

$G_{x|f} \leftarrow \texttt{DP-FT}(D^x_{\text{priv}}, D^f_{\text{priv}}, \sigma)$ ▷ Obtain an initial conditional text generator

// Step 3: Generate the best-of-N SFT dataset, and the feature dataset for RL

$D_{\text{SFT}_N} \leftarrow \texttt{Best-of-N-Sampling}(G_f, G_{x|f}, N, M_{\text{oracle}})$ ▷ Perform best-of-N sampling

$D^{\tilde{f}}_{\text{RL}} \leftarrow \texttt{Sampling}(G_f)$ ▷ Sample features as input to RL

// Step 4: Perform Anchored Reinforcement Learning

$G^{\text{ARL}}_{x|f} \leftarrow \texttt{AnchoredRL}(G_{x|f}, D^{\tilde{f}}_{\text{RL}}, D_{\text{SFT}_N})$ ▷ Train from $G_{x|f}$ using a weighted sum of losses

// Final step: Generate synthetic text

$D^{\tilde{f}}_{\text{syn}} \leftarrow \texttt{Sampling}(G_f)$ ▷ Sample synthetic features

$D^{\tilde{x}}_{\text{syn}} \leftarrow \texttt{Sampling}(G^{\text{ARL}}_{x|f}, D^{\tilde{f}}_{\text{syn}})$ ▷ Sample synthetic text conditioned on synthetic features

return $G_f, G^{\text{ARL}}_{x|f}, D^{\tilde{f}}_{\text{syn}}, D^{\tilde{x}}_{\text{syn}}$

4. **Anchored RL:** We fine-tune the initial generator $G_{x|f}$ using Anchored RL, which combines an RL objective on prompts from G_f with an SFT objective on the anchor dataset D_{SFT_N}. This leads to the final model $G^{\text{ARL}}_{x|f}$.

The procedure yields two key outputs: 1) a DP synthetic dataset, produced by sampling from G_f and $G^{\text{RL}}_{x|f}$, and 2) a conditional generator $G^{\text{ARL}}_{x|f}$ with good instruction-following capabilities, which can be used for on-demand, targeted generation tasks.

D.3 Additional Experimental Setups

D.3.1 Datasets

We adopt two challenging, real-world datasets for our studies.

bioRxiv [190] is a dataset of abstracts on the bioRxiv preprint server. The raw dataset is hosted on HuggingFace[38]. We filter the dataset to contain only the abstracts appearing

[38] `https://huggingface.co/datasets/hazylavender/biorxiv-abstract`

after the knowledge cutoff date of Gemma-3 family models (Aug 2024)[39]. We perform train/validation/test split and obtain a train set of size $n = 28,846$. We examine the token length for samples in the dataset, and found that the 95% quantile is 512 tokens. Thus we use context length 512 tokens with all methods.

PMC-patients [201] is a large-scale dataset of clinical notes documenting the patients' clinical visits. By nature, this dataset is highly sensitive. Upon examination, we found that only basic anonymity techniques were applied on the released dataset (e.g. redacting the patient's name). We use the latest dataset offered on HugggingFace[40] (version V2, released in 2024). We perform train/validation/test split and obtain a train set of size $n = 240,294$. We again use context length of 512 tokens.

D.3.2 Implementation Details for the Hierarchical Framework and Baselines

Aug-PE. We set the number of PE iterations T to 10 and number of variations L to 7 for both datasets, and perform privacy accounting using the code provided by the authors[41]. Following the recommendation in their paper [187], we perform selection by rank after the histogram voting.

DP-FT. We use the same code base of DP-FT, for all methods involving DP-FT as its subcomponents (vanilla DP-FT, as well as our conditional generation approaches). The codebase is adapted from Yu et al. [189][42]. Below we describe the hyperparameters: we use batch size $b = 2048$, iterations $T = 1120$ for bioRxiv (80 epochs) and $T = 1170$ for PMC-patients (10 epochs). We set clipping norm c to 1. We tune the learning rate $\eta \in$ {1e-4, 3e-4, 1e-3} and learning rate scheduler $\in$ {constant, cosine}. We perform LoRA fine-tuning [163] with a LoRA rank of 8, $\alpha = 16$ and dropout of 0.05. The same set of hyperparameters are used for vanilla DP-FT as well as DP-FT within our framework (for

[39] https://ai.google.dev/gemma/docs/core/model_card_3

[40] https://huggingface.co/datasets/zhengyun21/PMC-Patients

[41] https://github.com/AI-secure/aug-pe/blob/main/notebook/dp_budget.ipynb

[42] https://github.com/google-research/google-research/tree/master/dp_instructions/dp_finetuning

training G_f or $G_{x|f}$).

CTCL (as $\mathcal{S}_1$ in our framework). CTCL operates in the resource-constrained regime. We specifically adapt it to our setup: 1) Switch from an $O(100\text{M})$ encoder-decoder model to `gemma-3-1b-pt`, i.e., the same base model as used in our approaches. 2) Drop the pretraining stage in the original paper for the encoder-decoder model. The authors pretrained a topic model and released the model checkpoint on their GitHub[43]; we directly used the checkpoint for topic feature extraction. We follow their paper [191] to set the noise multiplier in the Gaussian mechanism (for the DP histogram) as $\sigma = 10$, and use $H = 0$ for thresholding the noisy histogram.

Sampling from a trained LLM. We perform nucleus decoding [248] to sample from trained LLMs and adopt the following hyper-paramters: temperature $T = 1.0$, top-$p = 0.95$, top-$k = 0$. We sample $N_{\text{syn}} = 5,000$ for all methods except for Aug-PE where we use $N_{\text{syn}} = 2,000$ following Xie et al. [187][44].

D.3.3 A Comprehensive Evaluation Suite

MAUVE. We follow Xie et al. [187] and Tan et al. [191] and use MAUVE to measure textual alignment. We use the code provided by Pillutla et al. [205][45].

- *Embedding model:* We identify several issues with the usage of embedding models in the current literature and provide detailed discussions in Appendix D.4.1. We choose specialized sentence embedding models that are suitable for each dataset. For bioRxiv, we choose `SPECTER2` [249][46] which is finetuned on abstracts of scientific papers. For PMC-patients, we adopt `S-PubMedBert-MS-MARCO` [250][47] which is finetuned on full-

[43] `https://github.com/tanyuqian/synthetic-private-data`

[44] This is due to the high cost of sampling in Aug-PE: T iterations where $N_{\text{syn}} \cdot (L-1)$ samples are generated in each iteration, plus $N_{\text{syn}} \cdot L$ samples generated in the first iteration.

[45] `https://github.com/krishnap25/mauve`

[46] `https://huggingface.co/allenai/specter2`

[47] `https://huggingface.co/pritamdeka/S-PubMedBert-MS-MARCO`

text PubMed articles. Both embedding models have context length (`max_seq_length`) of 512, same as the context length of the models we fine-tune,

- *Number of clusters*: We follow the recommendation of the authors[48] to use 1/10 of the synthetic data size as the number of clusters, i.e., 500 for evaluating all other methods, and 200 for evaluating Aug-PE generated samples. We additionally comment that when evaluating the same dataset, using a smaller number of clusters will lead to a higher MAUVE score; this is because the resulting cluster is coarser.

Classification F1. We craft a classification task to measure the utility of the synthetic data in the following way. We introduce a new attribute, and then use M_{oracle} to annotate both the real private test set and the generated data. After preparing the classification labels, we train on the synthetic data, and test on real data; this follows the standard evaluation approach considered in Yue et al. [185], Xie et al. [187], and Tan et al. [191].

- *Attribute and labels*: For bioRxiv, the new attribute is "research domain" with 8 meta-categories: {Biochemistry & Molecular Biology, Cell & Developmental Biology, Physiology & Immunology, Neuroscience & Cognition, Microbiology, Ecology & Evolution, Applied & Medical Biology, Computational Biology & Bioinformatics}.
- *Model*: We finetune a SciBERT model [251][49] for the classification task. We perform full fine-tuning.
- *Metric*: We use macro-F1 (unweighted average across classes), due to the class imbalance naturally present in the dataset (see Fig. 59).

Next token prediction (NTP) accuracy. We fine-tune a small LM on the synthetic data, and test on real private data. We follow the literature [187, 191] to fine-tune BERT-small and use the same set of standard hyperparameters.

[48] `https://krishnap25.github.io/mauve/`
[49] `https://huggingface.co/allenai/scibert_scivocab_uncased`

Attribute distribution matching. We compute **Jensen-Shannon distance** for each attribute separately and average over the attributes. The obtained score represents attribute distribution matching.

- *Jensen-Shannon divergence (JSD).* Given two probability distributions P and Q defined over the same domain, the Jensen-Shannon *divergence* (JSD) is defined as

$$\mathrm{JSD}(P \parallel Q) = \tfrac{1}{2}\,\mathrm{KL}(P \parallel M) + \tfrac{1}{2}\,\mathrm{KL}(Q \parallel M), \quad M = \tfrac{1}{2}(P+Q),$$

  where $\mathrm{KL}(P \parallel Q) = \sum_x P(x) \log \frac{P(x)}{Q(x)}$ denotes the Kullback-Leibler divergence. JSD is a smoothed version of KL that can handle potential support mismatch.

- *Jensen-Shannon distance* is given by the square root of the Jensen-Shannon divergence:

$$d_{\mathrm{JS}}(P, Q) := \sqrt{\mathrm{JSD}(P \parallel Q)}.$$

  This metric is symmetric, bounded in $[0, 1]$, and widely used to quantify similarity between probability distributions.

D.3.4 Implementation Details for ACTG-RL and ACTG-ARL

RL specific setups. We adopt TRL[50] as the RL fine-tuning framework and use the PPO objective (supported in `PPOTrainer`). We adapt the codebase from Singhal et al. [252][51] which provides an interface supporting different types of reward signals. For our ACTG-RL and ACTG-ARL, we implement the reward signal as the score from the LLM M_{oracle} grading on the rubric, and integrate it into the RL fine-tuning pipeline.

Training hyperparameters. We use a rollout buffer size of 512, batch size of 512 and ppo epochs of 4 (meaning we loop over the same rollout buffer for 4 times). We set the

[50] `https://huggingface.co/docs/trl/en/index`
[51] `https://github.com/PrasannS/rlhf-length-biases`

learning rate $\eta = 5 \times 10^{-6}$ and train for 1000 rounds in all. We set the initial KL coefficient to 0.2. For ACTG-ARL specifically, we introduce γ, a mixing coefficient for the RL and SFT objective. We adopt a linear decay schedule for γ, staring from a bigger γ for stabilizing the anchor and gradually decreasing it to allow for steady improvement in instruction following. For bioRxiv, we use a start value of 2 and ending value of 0.5. For PMC-patients, we use a start value of 0.5 and ending value of 0.2.

D.4 Additional Experimental Results

D.4.1 Issues with MAUVE Evaluation in the Literature

We highlight several critical issues with MAUVE evaluation in the literature [187, 191], which has gone unnoticed.

Limited context length. We report the default context length of common sequence embedding models (adopted in Xie et al. [187]) in Table 16. Note that all of the models have rather short context length (< 512). Because text beyond `max_seq_length` is truncated, these models can only evaluate quality with respect to a short prefix of the synthetic text. As a result, conclusions drawn from such biased evaluations may be significantly undermined.

Table 16: Default maximum sequence length for common sequence embedding models.

Embedding models in Xie et al. [187]	**Default context length** `max_seq_length`
`sentence-t5-xl`	256
`sentence-t5-base`	256
`stsb-roberta-base-v2`	75
`all-MiniLM-L6-v2`	256
`paraphrase-MiniLM-L6-v2`	128

Capability of the embedding model. We compute the MAUVE score of the same synthetic dataset w.r.t. embeddings extracted by different sequence embedding models. As

shown in Fig. 60, weaker embedding models (`stsb-roberta-base-v2`, `all-MiniLM-L6-v2`) can inflate the MAUVE score, while a stronger embedding model (`Qwen3-Embedding-4B` [253]) is more capable of assessing the quality of the synthetic data.

D.4.2 Baseline of Single-Stage Conditional Text Generation

A single-stage conditional generation approach. One straightforward approach that incorporates feature learning into text generation is to append features in front of text. Fig. 61 presents a template for bioRxiv. During *learning*, the model learns to generate the concatenation of feature and text following the template. During *generation*, we sample outputs from the model, and then discard the feature part and retain only the text part.

Experimental results. We compare the single-stage conditional generation approach with two approaches: 1) the baseline DP-FT—they are both single-stage approaches but differ in the conditioning part; 2) the two-stage conditional generation approach as we propose in Sec. 6.2.1—they both leverage conditioning but differ in the pipeline composition (single-stage vs. two-stage).

We conduct the experiments on bioRxiv and focus on $\mathcal{S}_2$. Results are presented in Fig. 62. **First**, the single-stage conditioning approach outperforms the baseline DP-FT at $\varepsilon = 1, 4$, meaning that conditioning is beneficial for DP text synthesis. **Second**, the two-stage conditional generation approach outperforms the single-stage one in both MAUVE and attribute distribution matching (left and middle). This demonstrates the *superiority of the two-stage conditioning generation framework*; even though the two-stage approach incurs privacy cost at both stages, the advantage of a better learning paradigm compensates for this downside. We conduct another sanity check to showcase the benefit of two-stage sequential conditional generation: we compute the MAUVE of features and show that the synthetic features produced by the two-stage approach is higher than that produced by the single-stage approach (Fig. 62(right)). This means the single-stage approach can first

produce better feature and then better text conditioned on the better feature.

D.4.3 Impact of Schema Richness

A simpler schema. For the bioRxiv dataset, we consider an alternative simpler schema $\mathcal{S}_3^{'}$. Unlike the $\mathcal{S}_3$ schema (Fig. 56) which consists of *eight* fields that need to be extracted by M_{oracle}, the simpler $\mathcal{S}_3^{'}$ schema contains only *three* ground-truth fields that are directly available or derivable from the dataset: `title`, `category` (the original data columns), and `token_count` (computed from the abstract). We present this schema in Fig. 63.

Instantiation of G_f and $G_{x|f}$. Since $\mathcal{S}_3^{'}$ contains textual features (e.g., `title`) that AIM cannot process, we adopt DP-FT as the feature generator G_f. For the conditional generator $G_{x|f}$, we employ the same DP-FT training procedure used for $\mathcal{S}_3$, learning on the paired (feature, text) set.

Results. Fig. 64 presents the end-to-end results on bioRxiv, where we add the new $\mathcal{S}_3^{'}$ (simple schema), as well as $\mathcal{S}_3$ used with DP-FT for feature generation for a fair comparison.

First, we compare $\mathcal{S}_3^{'}$ to the baselines. It significantly outperforms both vanilla DP-FT and CTCL. It also shows an advantage over the $\mathcal{S}_2$ (free-form) approach, indicating that even this simple schema provides a more effective conditioning signal than unstructured summaries.

Next, we directly assess the impact of schema richness by comparing $\mathcal{S}_3^{'}$ (simple schema) with the $\mathcal{S}_3$ (DP-FT+DP-FT) variant. The richer, 8-field $\mathcal{S}_3$ schema outperforms $\mathcal{S}_3^{'}$ in both MAUVE and feature distribution matching, confirming that a more comprehensive feature set is beneficial.

D.4.4 Experiments on a Larger Model `gemma-3-4b-pt`

In the main paper, we conducted systematic ablations using a single model, `gemma-3-1b-pt`. Here, we extend the evaluation to a larger model, **`gemma-3-4b-pt`**[52] , and show that the performance gains of ACTG persist at this larger scale.

Fig. 65 compares ACTG with the baselines (DP-FT and CTCL). The larger model improves the absolute performance of all methods, giving higher MAUVE scores and better distributional alignment compared to Fig. 24. ACTG nevertheless maintains a clear and substantial advantage over the baselines.

[52] `https://huggingface.co/google/gemma-3-4b-pt`

D.4.5 Robustness to Oracle Choice

Setup. To assess whether ACTG depends on a specific proprietary oracle, we replace `Gemini-2.5-flash-lite` with a locally deployed open-source model, `Qwen2.5-32B-Instruct`[53], using the same schema schema $\mathcal{S}_3$ (Fig. 56) with the same prompt template (Fig. 58) for feature extraction.

Feature-level comparison. We compute the agreement rate, defined as the ratio of matched fields per sample averaged across the dataset. The extracted features from Qwen and Gemini achieve an agreement rate of 0.69. As a reference, two independent Gemini runs achieve an agreement of 0.84. Because each field contains on average $\sim$13 categorical options and feature annotation could be inherently ambiguous, we do not expect perfect agreement. More importantly, an agreement of 0.69 already indicates that Qwen captures the core underlying semantics of these fields. This suggests that the extracted features are sufficiently aligned for our use of conditioning—we confirm this with our end-to-end results below.

End-to-end results. We further run the full ACTG pipeline on Qwen-extracted features and present the comparison with the results on Gemini-based pipeline in Fig. 66. The synthetic data produced by the Qwen-based pipeline attains nearly identical MAUVE scores and only minor degradation in feature-distribution matching (due to discrepancy in extracted feature distribution in training data).

These results show that ACTG is robust to the choice of feature extractor and can be effectively used with reasonably capable open-source models.

D.4.6 Significance of Results

We perform *three* independent runs with different random seeds for the downstream evaluation (classification F1) on bioRxiv. We report the mean and standard deviation for

[53] `https://huggingface.co/Qwen/Qwen2.5-32B-Instruct`

these runs in Fig. 67.

As the results in Fig. 67 show, the variance across runs is low for all methods. More importantly, the performance gap between ACTG and the baselines is substantially larger than the standard deviation. This analysis confirms that our reported gains on downstream tasks are robust and statistically significant.

D.4.7 Limitations of Direct Prompting as the Conditional Generator

Low MAUVE score. Fig. 68 shows that the MAUVE score of the direct prompting approach is extremely low. This is understandable as this approach does not have any access to the private text information, thus the poor textual alignment.

Inherent bias and distribution mismatch. Fig. 69-(left) shows that the oracle tends to overemphasize certain categories (e.g., Cell Biology), generating disproportionately more samples in these bins. Additionally, Fig. 69-(right) illustrates its inability to handle ambiguous or underspecified cases, leading to systematic errors when the input falls into categories such as "Other" or "Not Specified". These limitations stem from the fact that direct prompting has no means to calibrate feature distributions, as it handles each input feature independently. In contrast, our conditional generator obtained via DP-FT explicitly learns the mapping, enabling better attribute distribution matching.

D.4.8 Aug-PE on PMC-Patients

Aug-PE fails to produce meaningful results on PMC-patients. Even in the non-private setting, it achieves a MAUVE score below 0.05, attribute distribution matching d_{JS}^{f} higher than 0.2, and NTP accuracy of 0.32. Across all metrics, its performance lags far behind other methods. These negative results are likely caused by the large distribution shift between the PMC-patients dataset and the public corpus of pretraining.

D.4.9 Aug-PE with a More Powerful Proprietary Model

To further analyze the Aug-PE baseline, we perform an additional experiment with another powerful, state-of-the-art model, `Gemini-2.5-flash-lite`. Fig. 70 plots the performance of the model, compared with `Qwen2.5-7B-Instruct` across 10 PE iterations.

The results are striking: Aug-PE with `Qwen2.5-7B-Instruct` consistently and significantly outperforms Aug-PE with `Gemini-2.5-flash-lite` at both privacy levels. This finding strongly indicates that: the performance of PE is less dependent on the model's raw general-purpose capability and far more dependent on how well its initial "out-of-the-box" generations align with the private target distribution. As the figure shows, Qwen's initial population (iteration 0) is substantially better aligned with the private data, providing a strong starting point. Gemini, despite its capabilities, starts with a poorer alignment, and the PE process fails to close this significant gap. This confirms that simply using a "more powerful" model does not guarantee better PE performance; initial domain alignment is the critical factor.

D.4.10 Failure Mode of CTCL

Domain mismatch in topic extraction. CTCL relies on a pretrained topic model trained on general-domain corpora (specifically, Wikipedia). Such models can perform poorly when applied to narrow, domain-specific datasets, such as clinical notes (e.g., PMC-patients in our study). As illustrated in Fig. 71, a dental case is associated with unrelated keywords like "fossil" and "paleontology". Although these associations may arise from shared biological or anatomical terminology, they clearly fail to capture the intended clinical meaning. This mismatch underscores the sensitivity of topic-based extraction to distribution shift and its limited effectiveness in specialized domains.

Sparse DP histogram. When the number of samples is small relative to the number of topic bins (as in our bioRxiv dataset, where the dataset size is $N = 28{,}846$ but the number

of topics is 1,827), many bins can be empty. As all bins receive DP noise, when using a clipping threshold of zero (as in Tan et al. [191]), the noise can dominate. Consequently, as shown in Fig. 72, the DP histogram deviates significantly from the real distribution. This distortion harms both the fidelity and utility of synthetic text as reflected in Fig. 24 in Sec. 6.2.

D.4.11 Topic Distribution Matching

We further evaluate whether synthetic data preserves the topic distribution of the private dataset. We train a topic model using FASTopic [254][54] with $n_{\text{topics}} = 50$ on the *private training set*, ensuring that the model captures domain-specific topic structure. The trained model is then applied to both the private test set and the synthetic datasets to obtain their corresponding topic distributions. We then compute the Jensen-Shannon distance ($d_{\text{JS}}^{\text{topic}}$) between the two distributions.[55] As shown in Fig. 73, the benefits of attribute conditioning carry over to topic distribution alignment, with ACTG showing the closest match to the private data.

D.4.12 Using IT Model for Conditional Generation

We additionally experimented with using the instruction-tuned (IT) model `gemma-3-1b-it` as the base model for performing DP-FT to train $G_{x|f}$, instead of the pretrained (PT) model `gemma-3-1b-pt` which we have been using throughout the main paper. The intuition is that an IT model may already have stronger instruction-following ability, and thus can potentially achieve higher IFAcc even under DP. The below results are obtained at $\varepsilon = 1$ on bioRxiv.

[54] `https://github.com/bobxwu/FASTopic`

[55] Unlike the pre-trained topic model in Tan et al. [191] which was trained on general-domain corpora (Wikipedia), the FASTopic model here is trained directly on the private training set, providing the most faithful characterization of the private data, which is critical for this evaluation.

Overall comparison. Table 17 provides a quantitative comparison across multiple metrics. We found that the IT model performs poorer than the PT model across all metrics. We believe this stems from the following factors:

- *Objective mismatch.* IT models are tuned for instruction following rather than next-token modeling (as performed in DP-FT). Starting from an IT checkpoint gives higher perplexity on the target corpus and less headroom to improve under DP noise. We validate this in Fig. 74, showing that the IT model starts with a higher loss and plateaus at a much worse value than PT.
- *Generic helpfulness vs. domain alignment.* IT tuning bakes in a generic "helpful" style on public data. While this improves general instruction following on simple day-to-day tasks, it does not translate into better alignment with input features in our setting, which requires *domain-specific knowledge*.

Table 17: Comparison of `gemma-3-1b-pt` and `gemma-3-1b-it` as base models for performing DP-FT for conditional text generation.

	`gemma-3-1b-pt`	`gemma-3-1b-it`
MAUVE	**0.775**	0.419
IFAcc	**0.534**	0.503
d^f_{JS}	**0.087**	0.111
Classification F1	**0.726**	0.716

D.4.13 Example of Reward Hacking

Fig. 75 illustrates why the generated text in Fig. 26(c) receives a perfect score of 8/8 from M_{oracle}. Although the output is a very short TL;DR-style sentence, it explicitly satisfies every input field: the abstract mentions the correct research area, organism, data type, and focus scale, while also matching the expected approach, sample size, and research goal. Because each criterion is checked independently, the text achieves full credit despite lacking the length, detail, and stylistic fidelity of a proper scientific abstract. This example highlights how RL training can exploit the rubric reward, producing degenerate outputs that maximize score without preserving textual quality.

D.4.14 Analysis on Best-of-N Data

Fig. 76-(left) shows the maximum score difference across candidates for each prompt. Most prompts exhibit large variation, confirming that best-of-N has substantial room to improve over random sampling by consistently selecting the strongest candidate. Fig. 76-(right) reports per-rank IFAcc. The highest-ranked candidate (rank 1) achieves an average accuracy above 0.7, which is significantly higher than random samples. Together, these results validate best-of-N sampling as an effective way to distill a cleaner and higher-quality dataset without additional privacy cost.

D.4.15 Additional Ablations on Anchored RL

We further ablate the Anchored RL approach to disentangle the benefits of different design choices. In addition to ACTG-(A)RL, we introduce **ACTG-SFT**, where the conditional generator is fine-tuned directly on the anchor dataset (D_{SFT_1} or $D\mathrm{SFT}_N$) without reinforcement learning. This setup allows us to isolate the contribution of RL versus the quality of the anchor data itself.

RL vs. SFT. Fig. 77 shows that the ARL variants (orange) consistently outperform their SFT counterparts (green) in terms of IFAcc. This highlights the added value of reinforcement learning on anchored data: beyond what supervised fine-tuning alone can capture, RL further boosts fine-grained control.

Best-of-N sampling. Comparing the dark versus light hues, we see that models trained on D_{SFT_N} substantially outperform those trained on D_{SFT_1}. This confirms the importance of best-of-N sampling: using higher-quality anchors provides a much stronger training signal.

D.4.16 Utility Evaluation of Synthetic Data Produced by ACTG-ARL

Besides the evaluation of MAUVE and attribute distribution matching presented in Fig. 27 in Sec. 6.3.3, we additionally perform utility evaluation. Result in Fig. 78 shows that ACTG-ARL can further improve the utility of the generated synthetic data.

I would like to extract structured features from unstructured data. Your task is to analyze the dataset description and provided examples to define a set of representative categorical features.

 # Dataset Description

{data_description}

 # Primary Goal

The extracted features should be optimized to be as useful as possible for the following workload: {workload_description}

 # Core Task

Generate a set of {num_features} categorical features that provide a rich summary of the underlying text.

 # Feature Requirements:

1. **Feature Diversity**: The feature set should be comprehensive. Strive to include a mix of general-purpose features and domain-specific features.
2. **Orthogonality**: Prioritize features that are orthogonal / independent, unless they are intentionally hierarchical.
3. **Values**: Each feature must have a fixed set of at most 50 explicitly enumerated possible values. These values must be representative of the target data.
4. **Hierarchical Features**: Conditional features are permitted. If a feature's relevance depends on the value of another, its value should be "Not Applicable" when the condition is not met.

 # Output Format:

Provide your response as a numbered list. For each feature, you MUST include its name, possible values, a description, and a rationale for its inclusion.

1. **Feature Name**:
   - **Possible Values**: ...
   - **Description**: A brief, clear explanation of what the feature captures.
   - **Rationale**: A justification for why this feature is useful.

 # Examples:

{_formatted_examples}

Figure 55: A detailed prompt for schema identification. We fill in {data_description}, {workload_description}, {num_features} for each dataset based on general knowledge of the dataset domain. For {_formatted_examples}, this field is optional and we supply a few examples publicly available in the general domain.

{
"primary_research_area": "< Biochemistry | Bioinformatics | Biophysics | Cancer Biology | Cell Biology | Clinical Trials | Developmental Biology | Ecology | Epidemiology | Evolutionary Biology | Genetics | Genomics | Immunology | Microbiology | Molecular Biology | Neuroscience | Paleontology | Pathology | Pharmacology and Toxicology | Physiology | Plant Biology | Public Health | Scientific Communication and Education |Structural Biology | Synthetic Biology | Systems Biology | Zoology | Other>", // Categorizes the abstract into its main biological discipline.
"model_organism": "< Human | Mouse/Rat | Zebrafish | Drosophila melanogaster | Caenorhabditis elegans | Saccharomyces cerevisiae | Escherichia coli | Arabidopsis thaliana | Plant | Cell Culture | In Silico / Computational | Other Mammal | Other Vertebrate | Other Invertebrate | Other Microbe | Not Applicable / Review | Other >", // Identifies the primary biological model used in the research.
"experimental_approach": "< Wet Lab Experimentation | Computational / In Silico Analysis | Clinical Study | Field Study / Observation | Case Study / Case Review | Review / Meta-analysis | New Method Development | Theoretical Modeling | Other >", // Describes the main methodology used to conduct the study.
"dominant_data_type": "< Genomic | Transcriptomic | Proteomic | Metabolomic | Imaging | Structural | Phenotypic / Behavioral | Ecological / Environmental | Clinical / Patient Data | Simulation / Model Output | Multi-omics | Other >", // Specifies the primary type of data generated or analyzed in the paper.
"research_focus_scale": "<Molecular|Cellular|Circuit / Network|Tissue / Organ|Organismal|Population|Ecosystem|Multi-scale|Other>", // Categorizes the biological level of organization the study focuses on.
"disease_mention": "< Cancer | Neurodegenerative Disease | Infectious Disease | Metabolic Disease | Cardiovascular Disease | Autoimmune / Inflammatory Disease | Psychiatric / Neurological Disorder | Genetic Disorder | No Specific Disease Mentioned | Other >", // Identifies whether the abstract explicitly names a disease or a major disease category.
"sample_size": "< Single Subject / Case Study | Small Cohort (<50 subjects) | Medium Cohort (50-1000 subjects) | Large Cohort / Population-scale (>1000 subjects) | Relies on Cell/Animal Replicates | Not Specified / Not Applicable >", // Estimates the scale of the study based on mentions of sample or cohort size.
"research_goal": "< Investigating a mechanism | Characterizing a system/molecule | Developing a method/tool | Identifying novel elements | Testing a hypothesis | Quantifying a parameter | Evaluating/Comparing approaches | Other >" // Categorizes the study's primary objective based on its framing.
}

Figure 56: Schema for the bioRxiv dataset

{
"medical_specialty": "< Cardiology | Dermatology | Dentistry & Oral Surgery | Endocrinology | Gastroenterology | Hematology | Infectious Disease | Nephrology | Neurology | Obstetrics & Gynecology | Oncology | Ophthalmology | Orthopedics | Otolaryngology (ENT) | Pediatrics | Psychiatry | Pulmonology | Rheumatology | Surgery | Urology | Other >", // The primary medical discipline. Map sub-specialties (e.g., Neurosurgery) to their primary field.
"clinical_focus": "< Diagnostic Evaluation | Therapeutic Intervention | Monitoring & Follow-up | Adverse Event >", // The narrative's primary purpose (e.g., diagnosis, intervention, monitoring).
"patient_age_group": "< Neonate (0-28 days) | Pediatric (29 days-12 years) | Adolescent (13-17 years) | Adult (18-64 years) | Geriatric (65+ years) >", // The patient's specific age category.
"condition_chronicity": "< Acute | Chronic | Acute-on-Chronic | Recurrent / Relapsing | Congenital >", // The temporal nature and pattern of the patient's primary condition.
"narrative_structure": "< Chronological History | Problem-Oriented Summary | Procedural Report | Other >", // The organizational style and flow of the clinical summary.
"primary_intervention_type": "< Pharmacological | Surgical / Procedural | Supportive & Conservative Care | Not Applicable >", // The primary therapeutic or management action described (excluding diagnostic tests).
"diagnostic_certainty": "< Confirmed Diagnosis | Provisional Diagnosis | Differential Diagnosis | Not Applicable>", // The level of diagnostic confidence expressed within the narrative.
"patient_outcome": "< Resolved | Improved | Stable / Unchanged | Deteriorated | Deceased | Referred / Transferred | In-Progress / Unknown >" // The patient's clinical status or disposition at the end of the report.
}

Figure 57: Schema for the PMC-patients dataset

You are an expert biomedical information extraction assistant. Your task is to carefully read a scientific abstract from bioRxiv and extract the specified features according to the schema provided.

Output exactly one JSON object with no extra text or explanations.

CRITICAL INSTRUCTION 1: For any field in the JSON schema that lists specific options (e.g., "<Option1|Option2|...>"), you MUST select one of the provided options exactly as it is written. Do not invent, alter, or combine options. Failure to use an exact option from the list will be considered an error.

CRITICAL INSTRUCTION 2: Ensure the value chosen for a field is appropriate for that field's specific definition. Do not use an option from one field (e.g., 'Cellular' from 'research_focus_scale') as the value for another field.

Use this schema:
```json
{schema}
```

Abstract to analyze:

{abstract_text}

Your output (JSON only):

Figure 58: Prompt for feature extraction on the bioRxiv dataset, where {schema} is substituted with the content in Fig. 56 and {abstract_text} is the private text to be annotated.

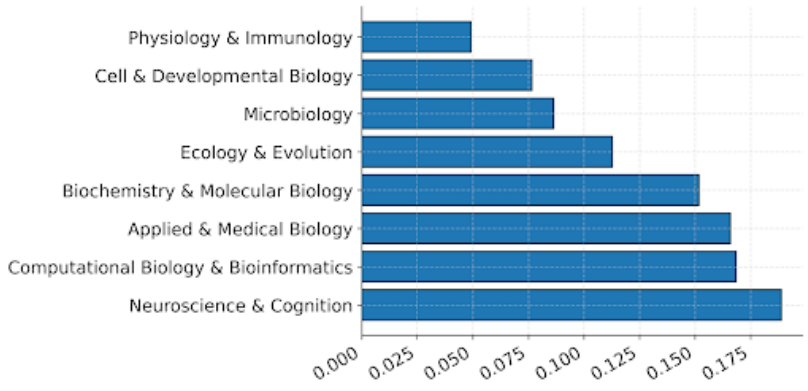


Figure 59: Histogram of labels of the "research domain" attribute in bioRxiv.

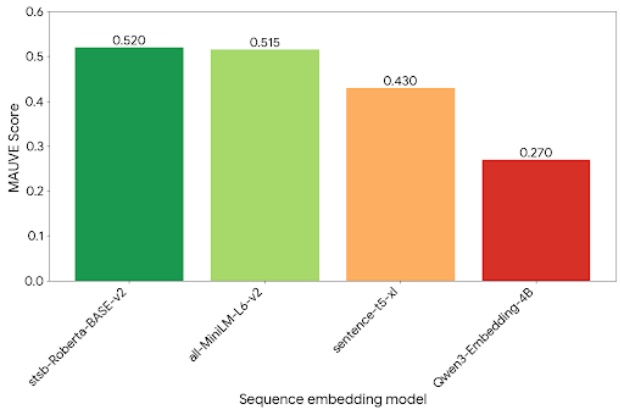


Figure 60: MAUVE score of the same synthetic dataset evaluated by different sequence embedding models.

```
##### SUMMARY #####
{summary}

##### ABSTRACT #####
{abstract}
```

Figure 61: A template for concatenating feature and text for bioRxiv.

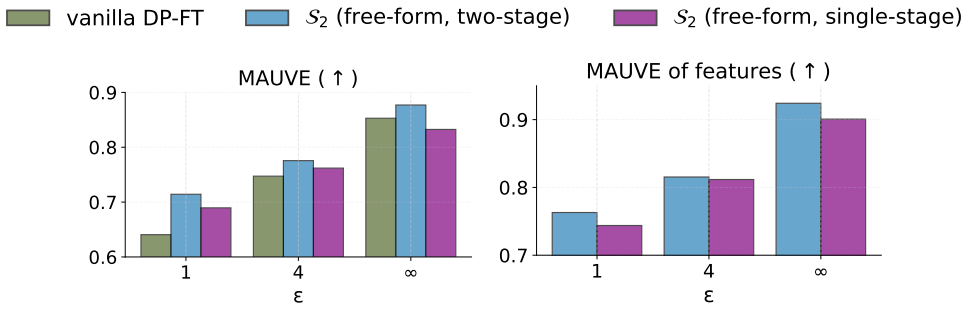


Figure 62: Comparisons of the single-stage conditional text generation with the baseline DP-FT and the two-stage conditional generation (our proposed framework). The left and middle figures show the comparison on MAUVE and attribute distribution matching. The right figure specifically shows the MAUVE of the synthetic *features*, comparing the single- and two-stage approaches.

```
{
  "title": "String", // Title of the paper.
  "category": "< bioengineering | cell biology | bioinformatics | synthetic biology | ecology | immunology | plant biology | cancer biology | developmental biology | microbiology | biophysics | genomics | biochemistry | evolutionary biology | pharmacology and toxicology | molecular biology | scientific communication and education | neuroscience | genetics | systems biology | physiology | zoology | animal behavior and cognition | pathology | paleontology >", // Category of the paper.
  "token_count": "Integer", // Number of tokens in the abstract.
}
```

Figure 63: A simpler 3-field schema $\mathcal{S}_3'$ for the bioRxiv dataset

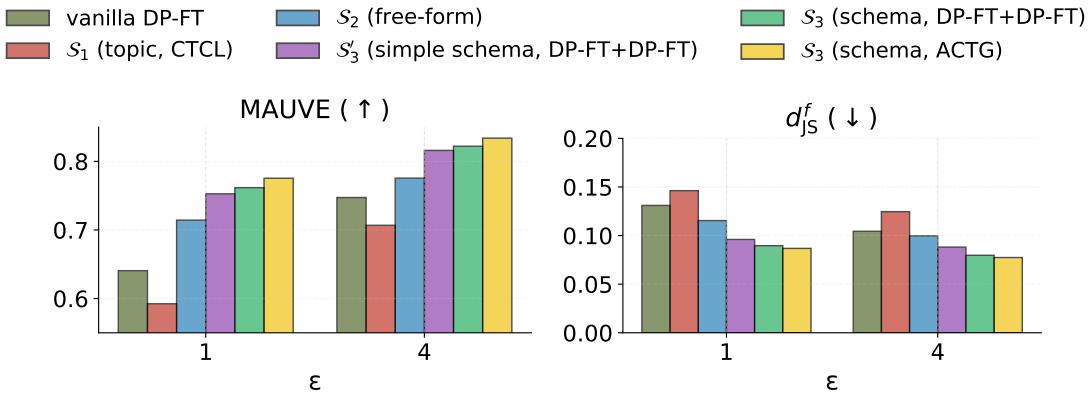


Figure 64: **Impact of schema richness.** We compare the simple 3-field schema ($\mathcal{S}_3'$) against baselines (DP-FT, CTCL), the free-form feature ($\mathcal{S}_2$), the rich 8-field $\mathcal{S}_3$ schema with a DP-FT feature generator, and the full ACTG ($\mathcal{S}_3$ with AIM). Results show that a richer schema outperforms a simple one, while the simpler one already offers clear advantages over the baselines.

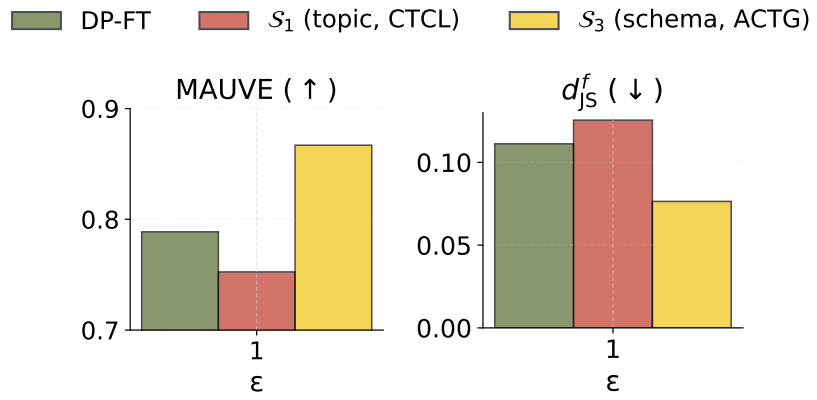


Figure 65: **Comparison of our ACTG with baselines (DP-FT, CTCL) on a larger model `gemma-3-4b-pt`** on bioRxiv, demonstrating its persistent performance advantage at scale.

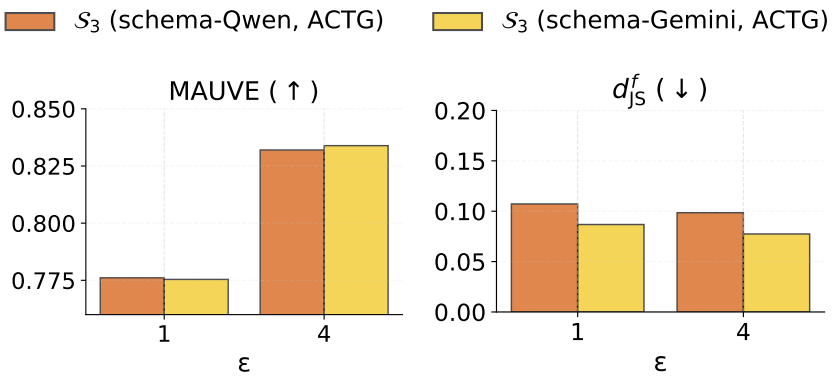


Figure 66: **End-to-end comparison of ACTG using features extracted by `Qwen2.5-32B-Instruct` and `Gemini-2.5-flash-lite`.** Dataset: bioRxiv. The Qwen-based pipeline achieves synthetic text quality that closely matches the Gemini-based pipeline, demonstrating robustness to the choice of feature extractor.

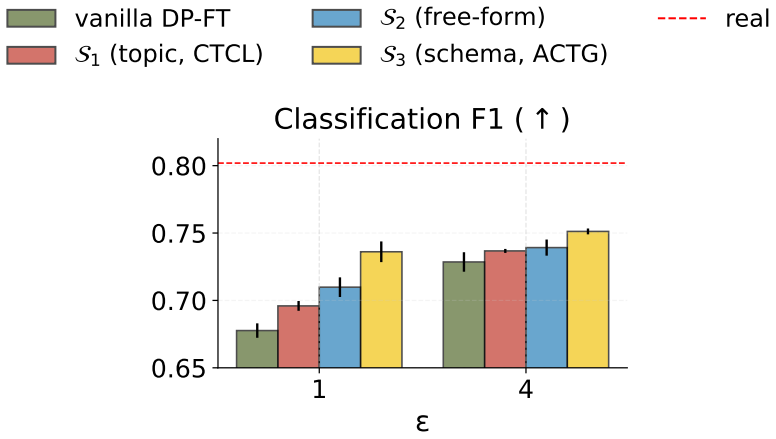


Figure 67: **Mean and standard deviation (black error bars) for the downstream evaluation** (classification F1 for bioRxiv) over *three* independent runs. The small variance and clear separation between ACTG and the baselines demonstrate the statistical significance of our gains.

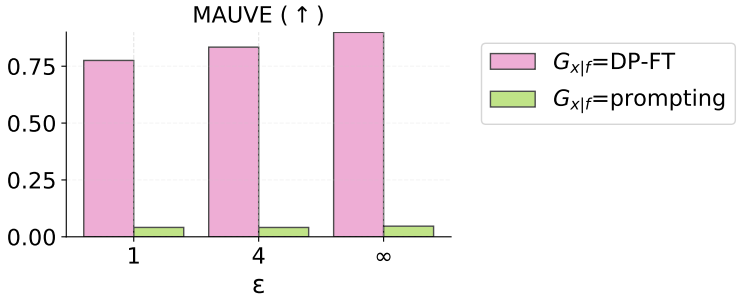


Figure 68: MAUVE scores achieved by different conditional generation approaches.

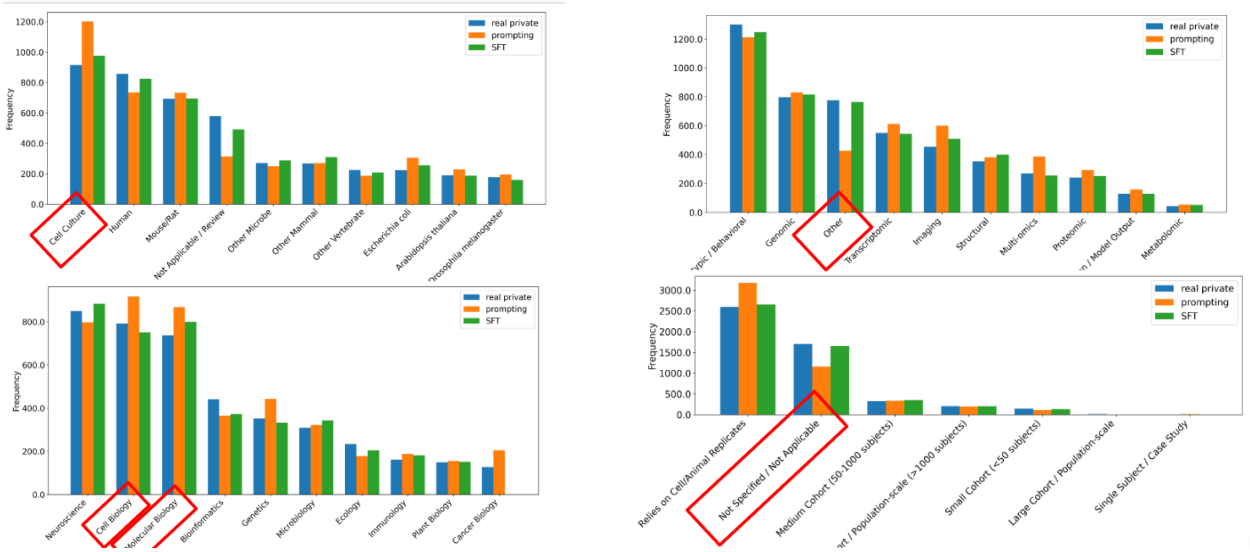


Figure 69: **(Left)** M_{oracle} favors concept related to Cell Biology and generates disproportionally more samples categorized to it. **(Right)** M_{oracle} fails to appropriately handle input of "Other" / "Not Specified".

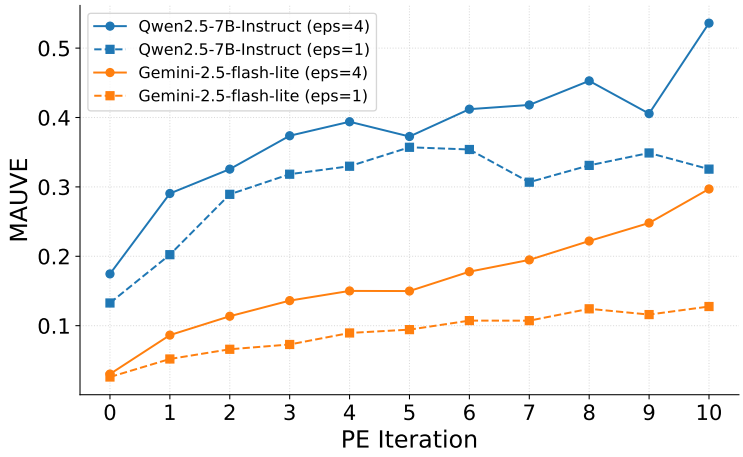


Figure 70: **Performance of Aug-PE using two different models: `Qwen2.5-7B-Instruct` and `Gemini-2.5-flash-lite`.** Dataset: bioRxiv. The gap highlights that PE's effectiveness critically depends on the alignment of the model's initial population with the target domain, not just its general capability.

Example Text: 'A 9-year-old healthy female child reported to our Outpatient Department with the chief complaint of swelling in the mandible for the past 6 months. As per the history, the swelling was painless and gradually increased to the present size. The patient underwent extraction of the left primary mandibular in the first molar for the same reason without any associated trauma, pain, or fever. Medical history was nonsignificant, with no history of systemic illness or long-term medication. On extraoral examination, there were no signs or symptoms. Intraorally, a diffuse swelling extending buccally from the distal of the primary mandibular left primary canine to the distal of the primary mandibular second primary molar ... (further text omitted)

Keywords associated with the predicted topic: 'fossil, paleontology, dinosaur, fossils, jurassic, phylogeny, cretaceous, phylogenetic, dinosaurs, prehistoric'

Figure 71: Example of spurious topic associations in CTCL. A clinical note for a dental visit is linked to keywords such as "fossil".

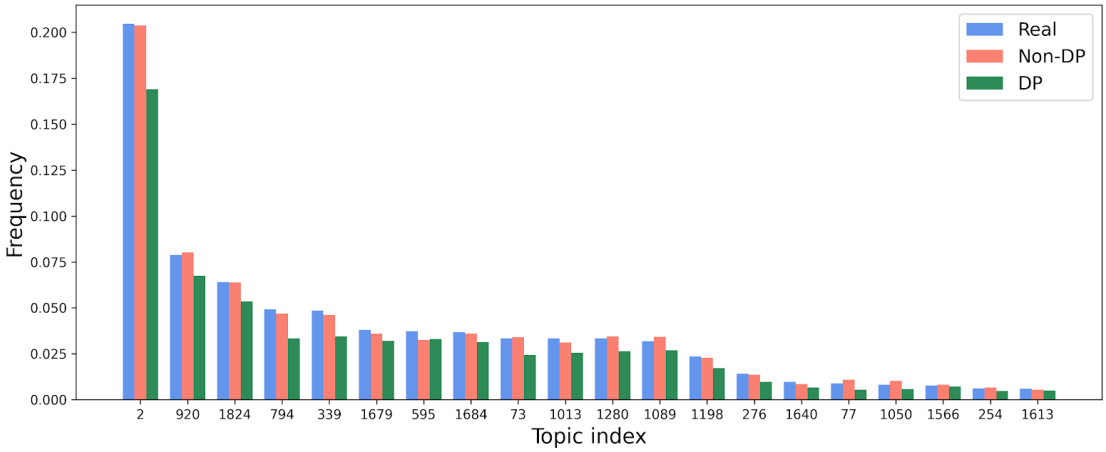


Figure 72: Comparison of topic histograms (top 10 topics, obtained on bioRxiv) of real data, non-DP samples, and DP samples. The non-DP histogram closely follows the real distribution, while the DP histogram diverges significantly due to noise amplification on sparse bins.

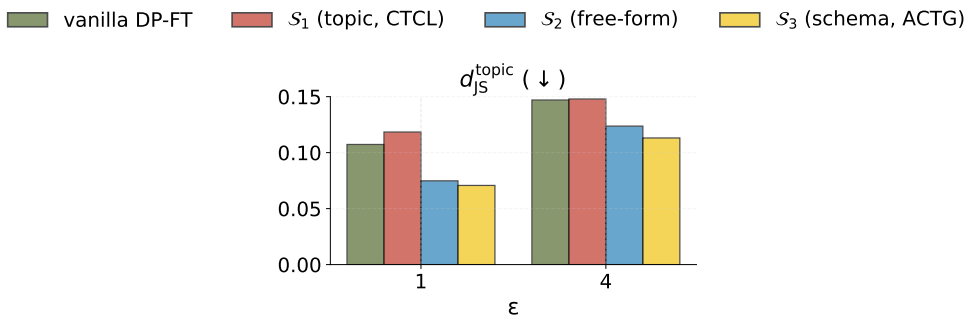


Figure 73: ACTG achieves the best topic distribution matching on bioRxiv.

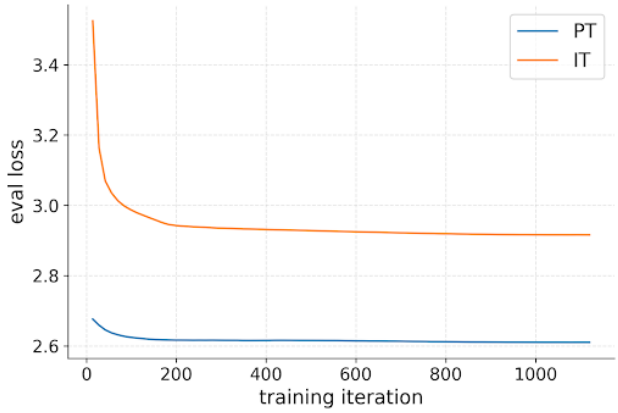


Figure 74: Evaluation loss during DP-FT on `gemma-3-1b-pt` and `gemma-3-1b-it`.

- `primary_research_area`: "Neuroscience" ✓ → The abstract mentions **"synaptic plasticity"** and **"spatial memory formation,"** which are core topics in the field of neuroscience.
- `model_organism`: "Drosophila melanogaster" ✓ → The abstract explicitly names the model organism, **"Drosophila."**
- `experimental_approach`: "Wet Lab Experimentation" ✓ → The phrase **"We experimentally evaluated"** directly reflects a hands-on, experimental approach consistent with wet lab work.
- `dominant_data_type`: "Phenotypic / Behavioral" ✓ → **"Spatial memory formation"** is a behavioral or phenotypic trait that is observed and measured in an organism.
- `research_focus_scale`: "Cellular" ✓ → **"Synaptic plasticity"** refers to the ability of synapses (the junctions between nerve cells) to strengthen or weaken over time, which is a phenomenon studied at the cellular level.
- `disease_mention`: "No Specific Disease Mentioned" ✓ → The abstract focuses on fundamental biological processes and does not mention any specific disease.
- `sample_size`: "Relies on Cell/Animal Replicates" ✓ → The study of **"Drosophila"** confirms the use of an animal model, which inherently relies on replicates for experimental validity.
- `research_goal`: "Investigating a mechanism" ✓ → The sentence structure, **"evaluated whether** [process A] **is preserved by modulating** [process B]," describes an investigation into the relationship between two processes, which is a form of investigating a mechanism.

Figure 75: Detailed breakdown of why the TL;DR-style generation in Fig. 26(c) receives a perfect score.

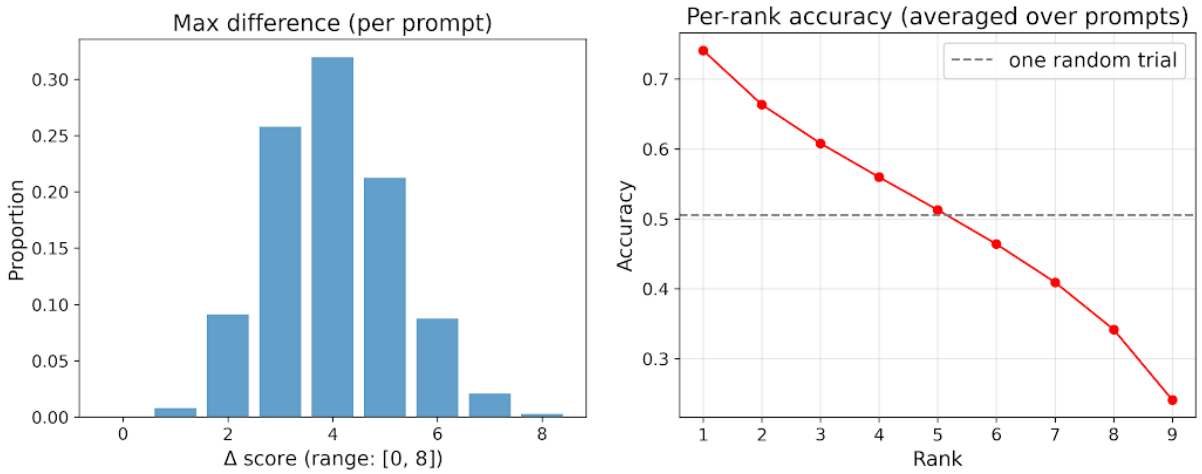


Figure 76: Analysis of best-of-N sampling. **(Left)** Distribution of max score difference per prompt, showing substantial room that best-of-N can exploit. **(Right)** Per-rank IFAcc, demonstrating that higher-ranked candidates can be significantly better than random samples.

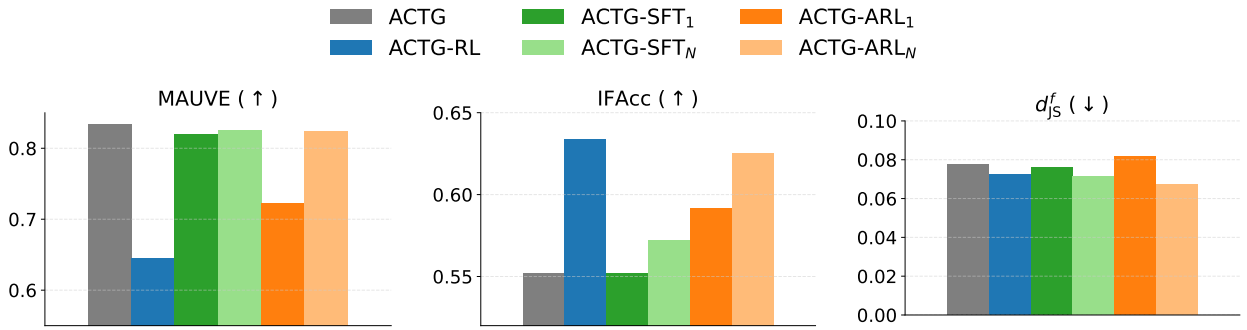


Figure 77: Ablation studies on Anchored RL, where we vary training data and training approaches.

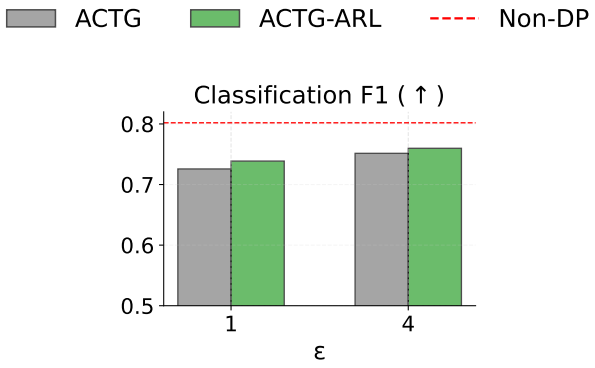


Figure 78: Utility evaluation of synthetic data produced by ACTG vs ACTG-ARL. Dataset: bioRxiv. We use the same Y-scale as in Fig. 24.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2012.

[2] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[3] Y. Zhu et al., "Aligning books and movies: Towards story-like visual explanations by watching movies and reading books," in *IEEE International Conference on Computer Vision*, 2015.

[4] *Common crawl*, `https://commoncrawl.org`, 2024.

[5] C. Raffel et al., "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of Machine Learning Research*, vol. 21, no. 140, pp. 1–67, 2020.

[6] J. Devlin, M.-W. Chang, K. Lee, and K. Tousignova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2019.

[7] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al., "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[8] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arber, S. von Arx, et al., "On the opportunities and risks of foundation models," *arXiv preprint arXiv:2108.07258*, 2021.

[9] J. Kaplan et al., "Scaling laws for neural language models," *arXiv preprint arXiv:2001.08361*, 2020.

[10] J. Hoffmann et al., "An empirical analysis of compute-optimal large language model training," in *Advances in Neural Information Processing Systems*, A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho, Eds., 2022. [Online]. Available: `https://openreview.net/forum?id=iBBcRUlOAPR`.

[11] Y. Bai et al., "Training a helpful and harmless assistant with reinforcement learning from human feedback," *arXiv preprint arXiv:2204.05862*, 2022.

[12] L. Ouyang et al., "Training language models to follow instructions with human feedback," *Advances in neural information processing systems*, vol. 35, pp. 27 730–27 744, 2022.

[13] Y. Lin, S. Zhang, B. Yu, J. Ye, J. Xu, et al., "Goedel-prover: A frontier model for open-source automated theorem proving," *arXiv preprint arXiv:2502.07640*, 2025.

[14] C. Schuhmann, R. Beaumont, R. Vencu, C. Gordon, R. Wightman, et al., "Laion-5b: An open large-scale dataset for training next generation image-text models," in *Advances in Neural Information Processing Systems*, 2022.

[15] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. L. Denton, et al., "Photorealistic text-to-image diffusion models with deep language understanding," in *Advances in Neural Information Processing Systems*, 2022.

[16] I. Shumailov, Z. Shumilo, Y. Zhao, N. Papernot, R. Anderson, and Y. Gal, "Ai models collapse when trained on recursively generated data," *Nature*, vol. 631, pp. 755–759, 2024.

[17] N. Carlini et al., "Extracting training data from large language models," in *30th USENIX Security Symposium (USENIX Security 21)*, 2021, pp. 2633–2650.

[18] N. Kandpal, E. Wallace, and C. Raffel, "Deduplicating training data mitigates privacy risks in language models," in *International Conference on Machine Learning*, 2022.

[19] X. Qi et al., "Fine-tuning aligned language models compromises safety, even when users do not intend to!" *arXiv preprint arXiv:2310.03693*, 2024.

[20] L. He, M. Xiong, and S. M. Xie, "What's in your "safe" data?: Identifying benign data that breaks safety," *arXiv preprint arXiv:2404.01099*, 2024.

[21] C. Zhou, P. Liu, P. Xu, S. Iyer, J. Sun, Y. Mao, et al., "Lima: Less is more for alignment," *arXiv preprint arXiv:2305.11206*, 2023.

[22] J. Li, A. Fang, G. Smyrnis, M. Ivgi, M. Jordan, S. Gadre, et al., "Datacomp-lm: In search of the next generation of training data for language models," *arXiv preprint arXiv:2406.11794*, 2024.

[23] Y. Ye, Z. Li, Y. Lin, Z. Lin, J. Wen, and J. Hao, "Limo: Less is more for reasoning," *arXiv preprint arXiv:2502.03387*, 2025.

[24] P. Maini, S. Saha, W. Zhuo, H. Yao, Y. Zhou, et al., "Rephrasing the web: A recipe for compute and data-efficient language modeling," *arXiv preprint arXiv:2401.16380*, 2025.

[25] P. Hu, Y. Hu, J. W. Ma, and H. Zhao, "A unified theory of random projection for influence functions," *arXiv preprint arXiv:2602.10449*, 2026.

[26] Y. Hu et al., "A snapshot of influence: A local data attribution framework for online reinforcement learning," in *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. [Online]. Available: `https://openreview.net/forum?id=sYK4yPDuT1`.

[27] Y. Hu et al., "Empirical privacy variance," in *Forty-second International Conference on Machine Learning*, 2025. [Online]. Available: `https://openreview.net/forum?id=oEvbe7vtOm`.

[28] Y. Hu et al., "Actg-arl: Differentially private conditional text generation with rl-boosted control," *arXiv preprint arXiv:2510.18232*, 2025.

[29] J. Deng et al., "A survey of data attribution: Methods, applications, and evaluation in the era of generative ai," *SSRN*, 2025, Available at SSRN: https://ssrn.com/abstract=5451054. DOI: 10.2139/ssrn.5451054.

[30] Y. Hu, P. Hu, H. Zhao, and J. Ma, "Most influential subset selection: Challenges, promises, and beyond," in *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. [Online]. Available: https://openreview.net/forum?id=qWi33pPecC.

[31] Y. Hu, R. Xian, Q. Wu, Q. Fan, L. Yin, and H. Zhao, "Revisiting scalarization in multi-task learning: A theoretical perspective," in *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. [Online]. Available: https://openreview.net/forum?id=6EqUpqMnwl.

[32] Y. Hu, F. Wu, H. Zhang, and H. Zhao, "Understanding the impact of adversarial robustness on accuracy disparity," in *International Conference on Machine Learning*, PMLR, 2023, pp. 13 679–13 709.

[33] Y. Hu et al., "Sok: Privacy-preserving data synthesis," in *2024 IEEE Symposium on Security and Privacy (SP)*, IEEE, 2024, pp. 4696–4713.

[34] Z. Hammoudeh and D. Lowd, "Training data influence analysis and estimation: A survey," *Machine Learning*, pp. 1–53, 2024.

[35] F. R. Hampel, "The influence curve and its role in robust estimation," *Journal of the american statistical association*, vol. 69, no. 346, pp. 383–393, 1974.

[36] P. W. Koh and P. Liang, "Understanding black-box predictions via influence functions," in *International conference on machine learning*, PMLR, 2017, pp. 1885–1894.

[37] M. Wojnowicz et al., ""influence sketching": Finding influential samples in large-scale regressions," in *2016 IEEE International Conference on Big Data (Big Data)*, IEEE, 2016, pp. 3601–3612.

[38] S. M. Park, K. Georgiev, A. Ilyas, G. Leclerc, and A. Madry, "Trak: Attributing model behavior at scale," in *International Conference on Machine Learning*, PMLR, 2023, pp. 27 074–27 113.

[39] S. K. Choe et al., "What is your data worth to gpt? llm-scale data valuation with influence functions," *arXiv preprint arXiv:2405.13954*, 2024.

[40] G. Pruthi, F. Liu, S. Kale, and M. Sundararajan, "Estimating training data influence by tracing gradient descent," *Advances in Neural Information Processing Systems*, vol. 33, pp. 19 920–19 930, 2020.

[41] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Conference on Theory of Cryptography*, TCC '06, 2006.

[42] M. Abadi et al., "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, 2016, pp. 308–318.

[43] S. Gopi, Y. T. Lee, and L. Wutschitz, "Numerical composition of differential privacy," *Advances in Neural Information Processing Systems*, vol. 34, pp. 11 631–11 642, 2021.

[44] V. Doroshenko, B. Ghazi, P. Kamath, R. Kumar, and P. Manurangsi, "Connect the dots: Tighter discrete approximations of privacy loss distributions," *Proceedings on Privacy Enhancing Technologies*, 2022.

[45] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *The Third International Conference on Learning Representations*, 2015.

[46] X. Li, F. Tramer, P. Liang, and T. Hashimoto, "Large language models can be strong differentially private learners," in *International Conference on Learning Representations*, 2022. [Online]. Available: `https://openreview.net/forum?id=bVuP3ltATMz`.

[47] S. De, L. Berrada, J. Hayes, S. L. Smith, and B. Balle, "Unlocking high-accuracy differentially private image classification through scale," *arXiv preprint arXiv:2204.13650*, 2022.

[48] D. Yu et al., "Differentially private fine-tuning of language models," in *International Conference on Learning Representations*, 2022. [Online]. Available: `https://openreview.net/forum?id=Q42f0dfjECO`.

[49] Z. Xu et al., "Federated learning of gboard language models with differential privacy," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 5: Industry Track)*, 2023, pp. 629–639.

[50] P. Hu, J. Melkonian, W. Tang, H. Zhao, and J. W. Ma, "Grass: Scalable data attribution with gradient sparsification and sparse projection," in *Advances in Neural Information Processing Systems*, 2025.

[51] W. J. J. Lindenstrauss and J. Johnson, "Extensions of lipschitz maps into a hilbert space," *Contemp. Math*, vol. 26, no. 189-206, p. 2, 1984.

[52] N. Ailon and B. Chazelle, "The fast johnson–lindenstrauss transform and approximate nearest neighbors," *SIAM Journal on computing*, vol. 39, no. 1, pp. 302–322, 2009.

[53] D. M. Kane and J. Nelson, "Sparser johnson-lindenstrauss transforms," *Journal of the ACM (JACM)*, vol. 61, no. 1, pp. 1–23, 2014.

[54] J. Nelson and H. L. Nguyên, "Osnap: Faster numerical linear algebra algorithms via sparser subspace embeddings," in *2013 ieee 54th annual symposium on foundations of computer science*, IEEE, 2013, pp. 117–126.

[55] M. B. Cohen, "Nearly tight oblivious subspace embeddings by trace inequalities," in *Proceedings of the twenty-seventh annual ACM-SIAM symposium on Discrete algorithms*, SIAM, 2016, pp. 278–287.

[56] W. Wang et al., "Taming hyperparameter sensitivity in data attribution: Practical selection without costly retraining," in *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. [Online]. Available: https://openreview.net/forum?id=qVDEM93mCP.

[57] X. Zheng, T. Pang, C. Du, J. Jiang, and M. Lin, "Intriguing properties of data attribution on diffusion models," in *The Twelfth International Conference on Learning Representations*, 2024.

[58] B. K. Mlodozeniec, R. Eschenhagen, J. Bae, A. Immer, D. Krueger, and R. E. Turner, "Influence functions for scalable data attribution in diffusion models," in *The Thirteenth International Conference on Learning Representations*, 2025. [Online]. Available: https://openreview.net/forum?id=esYrEndGsr.

[59] J. Martens and R. Grosse, "Optimizing neural networks with kronecker-factored approximate curvature," in *Proceedings of the 32nd International Conference on Machine Learning*, F. Bach and D. Blei, Eds., ser. Proceedings of Machine Learning Research, vol. 37, Lille, France: PMLR, Jul. 2015, pp. 2408–2417. [Online]. Available: https://proceedings.mlr.press/v37/martens15.html.

[60] T. George, C. Laurent, X. Bouthillier, N. Ballas, and P. Vincent, "Fast approximate natural gradient descent in a kronecker factored eigenbasis," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[61] J. Bae, N. H. Ng, A. Lo, M. Ghassemi, and R. B. Grosse, "If influence functions are the answer, then what is the question?" In *Advances in Neural Information Processing Systems*, A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho, Eds., 2022.

[62] R. Grosse et al., *Studying large language model generalization with influence functions*, 2023. arXiv: 2308.03296 [cs.LG].

[63] Y. Kwon, E. Wu, K. Wu, and J. Zou, "Datainf: Efficiently estimating data influence in loRA-tuned LLMs and diffusion models," in *The Twelfth International Conference*

on Learning Representations, 2024. [Online]. Available: https://openreview.net/forum?id=9m02ib92Wz.

[64] R. Vershynin, *High-dimensional probability: An introduction with applications in data science*. Cambridge university press, 2018, vol. 47.

[65] S. F. Gull, "Developments in maximum entropy data analysis," in *Maximum Entropy and Bayesian Methods: Cambridge, England, 1988*, Springer, 1989, pp. 53–71.

[66] D. MacKay, "Bayesian model comparison and backprop nets," *Advances in neural information processing systems*, vol. 4, 1991.

[67] D. P. Woodruff, "Sketching as a tool for numerical linear algebra," *Foundations and Trends® in Theoretical Computer Science*, vol. 10, no. 1–2, pp. 1–157, 2014.

[68] J. Deng et al., "Dattri: A library for efficient data attribution," in *Advances in Neural Information Processing Systems*, A. Globerson et al., Eds., vol. 37, Curran Associates, Inc., 2024, pp. 136 763–136 781. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2024/file/f732683302d91e47610b2416b4977a66-Paper-Datasets_and_Benchmarks_Track.pdf.

[69] S. Teso, A. Bontempelli, F. Giunchiglia, and A. Passerini, "Interactive label cleaning with example-based explanations," in *Advances in Neural Information Processing Systems*, 2021.

[70] H. Guo, N. Rajani, P. Hase, M. Bansal, and C. Xiong, "FastIF: Scalable influence functions for efficient model interpretation and debugging," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Nov. 2021, pp. 10 333–10 350.

[71] A. Schioppa, P. Zablotskaia, D. Vilar, and A. Sokolov, "Scaling up influence functions," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 8, pp. 8179–8186, Jun. 2022.

[72] A. Schioppa, "Efficient sketches for training data attribution and studying the loss landscape," *Advances in Neural Information Processing Systems*, vol. 37, pp. 37 692–37 735, 2024.

[73] V. Mnih et al., "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[74] D. Silver et al., "Mastering the game of go with deep neural networks and tree search," *nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[75] O. M. Andrychowicz et al., "Learning dexterous in-hand manipulation," *The International Journal of Robotics Research*, vol. 39, no. 1, pp. 3–20, 2020.

[76] V. Mnih et al., "Asynchronous methods for deep reinforcement learning," in *International conference on machine learning*, PmLR, 2016, pp. 1928–1937.

[77] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[78] A. E. Sallab, M. Abdou, E. Perot, and S. Yogamani, "Deep reinforcement learning framework for autonomous driving," *arXiv preprint arXiv:1704.02532*, 2017.

[79] P. Henderson, R. Islam, P. Bachman, J. Pineau, D. Precup, and D. Meger, "Deep reinforcement learning that matters," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, 2018.

[80] Y. Yu, "Towards sample efficient reinforcement learning.," in *IJCAI*, 2018, pp. 5739–5743.

[81] G. Dulac-Arnold, D. Mankowitz, and T. Hester, *Challenges of real-world reinforcement learning*, 2019. [Online]. Available: `https://openreview.net/forum?id=S1xtR52NjN`.

[82] S. Milani, N. Topin, M. Veloso, and F. Fang, "Explainable reinforcement learning: A survey and comparative review," *ACM Computing Surveys*, vol. 56, no. 7, pp. 1–36, 2024.

[83] Z. Cheng, J. Yu, and X. Xing, "A survey on explainable deep reinforcement learning," *arXiv preprint arXiv:2502.06869*, 2025.

[84] M. Xia, S. Malladi, S. Gururangan, S. Arora, and D. Chen, "LESS: Selecting influential data for targeted instruction tuning," in *Forty-first International Conference on Machine Learning*, 2024. [Online]. Available: `https://openreview.net/forum?id=PG5fV50maR`.

[85] H. Wang, Z. Wu, and J. He, "Fairif: Boosting fairness in deep learning via influence functions with validation set sensitive attributes," in *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, 2024, pp. 721–730.

[86] T. A. Chang, D. Rajagopal, T. Bolukbasi, L. Dixon, and I. Tenney, "Scalable influence and fact tracing for large language model pretraining," in *The Thirteenth International Conference on Learning Representations*, 2025. [Online]. Available: `https://openreview.net/forum?id=gLa96FlWwn`.

[87] C. Berner et al., "Dota 2 with large scale deep reinforcement learning," *arXiv preprint arXiv:1912.06680*, 2019.

[88] T. Xie, H. Li, A. Bai, and C.-J. Hsieh, "Data attribution for diffusion models: Timestep-induced bias in influence estimation," *Transactions on Machine Learning Research*, 2024, ISSN: 2835-8856. [Online]. Available: `https://openreview.net/forum?id=P3Lyun7CZs`.

[89] H. Lin, J. Long, Z. Xu, and W. Zhao, "Token-wise influential training data retrieval for large language models," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2024, pp. 841–860.

[90] A. Ghorbani and J. Zou, "Data shapley: Equitable valuation of data for machine learning," in *International conference on machine learning*, PMLR, 2019, pp. 2242–2251.

[91] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, Second. Cambridge, MA: The MIT Press, 2018. [Online]. Available: `http://incompleteideas.net/book/the-book-2nd.html`.

[92] Z. Shao et al., "Deepseekmath: Pushing the limits of mathematical reasoning in open language models," *arXiv preprint arXiv:2402.03300*, 2024.

[93] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *International conference on machine learning*, PMLR, 2015, pp. 1889–1897.

[94] V. Mnih et al., "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[95] A. Ilyas et al., "A closer look at deep policy gradients," *arXiv preprint arXiv:1811.02553*, 2018.

[96] C. Spearman, "The proof and measurement of association between two things," *The American Journal of Psychology*, vol. 15, no. 1, pp. 72–101, 1904.

[97] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and computing*, vol. 17, pp. 395–416, 2007.

[98] F. R. Chung, *Spectral graph theory*. American Mathematical Soc., 1997, vol. 92.

[99] J. T. Wang, P. Mittal, D. Song, and R. Jia, "Data shapley in one training run," in *The Thirteenth International Conference on Learning Representations*, 2025. [Online]. Available: `https://openreview.net/forum?id=HD6bWcj87Y`.

[100] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized experience replay," in *International Conference on Learning Representations (ICLR)*, 2016. arXiv: 1511.05952. [Online]. Available: http://arxiv.org/abs/1511.05952.

[101] W. Muldrew, P. Hayes, M. Zhang, and D. Barber, "Active preference learning for large language models," in *Forty-first International Conference on Machine Learning*, 2024. [Online]. Available: https://openreview.net/forum?id=CTgEV6qgUy.

[102] N. Das, S. Chakraborty, A. Pacchiano, and S. R. Chowdhury, "Active preference optimization for sample efficient RLHF," in *ICML 2024 Workshop on Theoretical Foundations of Foundation Models*, 2024. [Online]. Available: https://openreview.net/forum?id=uSCvfYNn0s.

[103] Y. Shen, H. Sun, and J.-F. Ton, "Reviving the classics: Active reward modeling in large language model alignment," *arXiv preprint arXiv:2502.04354*, 2025.

[104] Hugging Face, *Detoxifying a language model using ppo*, https://huggingface.co/docs/trl/en/detoxifying_a_lm, TRL documentation (v0.17.0), accessed May 8, 2025, 2023.

[105] S. Black, G. Leo, P. Wang, C. Leahy, and S. Biderman, *GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow*, version 1.0, If you use this software, please cite it using these metadata., Mar. 2021. DOI: 10.5281/zenodo.5297715. [Online]. Available: https://doi.org/10.5281/zenodo.5297715.

[106] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep reinforcement learning: A brief survey," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 26–38, 2017.

[107] T. Zahavy, N. Ben-Zrihem, and S. Mannor, "Graying the black box: Understanding dqns," in *International conference on machine learning*, PMLR, 2016, pp. 1899–1908.

[108] S. Greydanus, A. Koul, J. Dodge, and A. Fern, "Visualizing and understanding atari agents," in *International conference on machine learning*, PMLR, 2018, pp. 1792–1801.

[109] A. Mott, D. Zoran, M. Chrzanowski, D. Wierstra, and D. Jimenez Rezende, "Towards interpretable reinforcement learning using attention augmented agents," *Advances in neural information processing systems*, vol. 32, 2019.

[110] A. Atrey, K. Clary, and D. Jensen, "Exploratory not explanatory: Counterfactual analysis of saliency maps for deep reinforcement learning," in *International Conference on Learning Representations*, 2020. [Online]. Available: `https://openreview.net/forum?id=rkl3m1BFDB`.

[111] N. Puri et al., "Explain your move: Understanding agent actions using specific and relevant feature attribution," in *International Conference on Learning Representations*, 2020. [Online]. Available: `https://openreview.net/forum?id=SJgzLkBKPB`.

[112] A. Verma, V. Murali, R. Singh, P. Kohli, and S. Chaudhuri, "Programmatically interpretable reinforcement learning," in *International conference on machine learning*, PMLR, 2018, pp. 5045–5054.

[113] E. Soares, P. P. Angelov, B. Costa, M. P. G. Castro, S. Nageshrao, and D. Filev, "Explaining deep learning models through rule-based approximation and visualization," *IEEE Transactions on Fuzzy Systems*, vol. 29, no. 8, pp. 2399–2407, 2020.

[114] N. Topin, S. Milani, F. Fang, and M. Veloso, "Iterative bounding mdps: Learning interpretable policies via non-interpretable methods," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, 2021, pp. 9923–9931.

[115] C. Demircan, T. Saanum, A. K. Jagadish, M. Binz, and E. Schulz, "Sparse autoencoders reveal temporal difference learning in large language models," in *The Thirteenth International Conference on Learning Representations*, 2025. [Online]. Available: `https://openreview.net/forum?id=2tIyA5cri8`.

[116] Z. Juozapaitis, A. Koul, A. Fern, M. Erwig, and F. Doshi-Velez, "Explainable reinforcement learning via reward decomposition," in *IJCAI/ECAI Workshop on explainable artificial intelligence*, 2019.

[117] S. Liu and M. Zhu, "UTILITY: Utilizing explainable reinforcement learning to improve reinforcement learning," in *The Thirteenth International Conference on Learning Representations*, 2025. [Online]. Available: `https://openreview.net/forum?id=Tk1VQDadfL`.

[118] S. V. Deshmukh et al., "Explaining RL decisions with trajectories," in *The Eleventh International Conference on Learning Representations*, 2023. [Online]. Available: `https://openreview.net/forum?id=5Egggz1q575`.

[119] W. Guo, X. Wu, U. Khan, and X. Xing, "Edge: Explaining deep reinforcement learning policies," *Advances in Neural Information Processing Systems*, vol. 34, pp. 12 222–12 236, 2021.

[120] J. Yu, W. Guo, Q. Qin, G. Wang, T. Wang, and X. Xing, "{Airs}: Explanation for deep reinforcement learning based security applications," in *32nd USENIX Security Symposium (USENIX Security 23)*, 2023, pp. 7375–7392.

[121] H. Liu et al., "Learning to identify critical states for reinforcement learning from videos," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 1955–1965.

[122] R. Rishav, S. Nath, V. Michalski, and S. E. Kahou, "Behaviour discovery and attribution for explainable reinforcement learning," *arXiv preprint arXiv:2503.14973*, 2025.

[123] A. Jacq, J. Ferret, O. Pietquin, and M. Geist, "Lazy-mdps: Towards interpretable rl by learning when to act," in *Proceedings of the International Foundation for Autonomous Agents and Multiagent Systems*, 2022, pp. 669–677.

[124] Z. Cheng, X. Wu, J. Yu, W. Sun, W. Guo, and X. Xing, "Statemask: Explaining deep reinforcement learning through state mask," *Advances in Neural Information Processing Systems*, vol. 36, pp. 62 457–62 487, 2023.

[125] Z. Cheng, X. Wu, J. Yu, S. Yang, G. Wang, and X. Xing, "RICE: Breaking through the training bottlenecks of reinforcement learning with explanation," in *Forty-first International Conference on Machine Learning*, 2024. [Online]. Available: `https://openreview.net/forum?id=PKJqsZD5nQ`.

[126] N. Stiennon et al., "Learning to summarize with human feedback," *Advances in Neural Information Processing Systems*, vol. 33, pp. 3008–3021, 2020.

[127] X. Wang et al., "Self-consistency improves chain of thought reasoning in language models," in *The Eleventh International Conference on Learning Representations*, 2023. [Online]. Available: `https://openreview.net/forum?id=1PL1NIMMrw`.

[128] B. Roziere et al., "Code llama: Open foundation models for code," *arXiv preprint arXiv:2308.12950*, 2023.

[129] A. Dubey et al., "The llama 3 herd of models," *arXiv preprint arXiv:2407.21783*, 2024.

[130] Gemma Team et al., "Gemma 2: Improving open language models at a practical size," *arXiv preprint arXiv:2408.00118*, 2024.

[131] A. Liu et al., "Deepseek-v3 technical report," *arXiv preprint arXiv:2412.19437*, 2024.

[132] X. Guo and H. Yu, "On the domain adaptation and generalization of pretrained language models: A survey," *arXiv preprint arXiv:2211.03154*, 2022.

[133] Y. Li et al., "Personal llm agents: Insights and survey about the capability, efficiency and security," *arXiv preprint arXiv:2401.05459*, 2024.

[134] N. Carlini, D. Ippolito, M. Jagielski, K. Lee, F. Tramer, and C. Zhang, "Quantifying memorization across neural language models," in *The Eleventh International Conference on Learning Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=TatRHT_1cK.

[135] N. Lukas, A. Salem, R. Sim, S. Tople, L. Wutschitz, and S. Zanella-Béguelin, "Analyzing leakage of personally identifiable information in language models," in *IEEE Symposium on Security and Privacy (SP)*, 2023.

[136] S. Biderman et al., "Emergent and predictable memorization in large language models," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[137] U. S. Prashanth et al., "Recite, reconstruct, recollect: Memorization in LMs as a multifaceted phenomenon," in *The Thirteenth International Conference on Learning Representations*, 2025. [Online]. Available: https://openreview.net/forum?id=3E8YNv1HjU.

[138] R. Anil, B. Ghazi, V. Gupta, R. Kumar, and P. Manurangsi, "Large-scale differentially private bert," *arXiv preprint arXiv:2108.01624*, 2021.

[139] Z. Bu, Y.-X. Wang, S. Zha, and G. Karypis, "Differentially private optimization on large model at small cost," in *International Conference on Machine Learning*, PMLR, 2023, pp. 3192–3218.

[140] F. Wu, H. A. Inan, A. Backurs, V. Chandrasekaran, J. Kulkarni, and R. Sim, "Privately aligning language models with reinforcement learning," in *The Twelfth International Conference on Learning Representations*, 2024. [Online]. Available: https://openreview.net/forum?id=3d0OmYTNui.

[141] H. Brown, K. Lee, F. Mireshghallah, R. Shokri, and F. Tramèr, "What does it mean for a language model to preserve privacy?" In *Proceedings of the 2022 ACM conference on fairness, accountability, and transparency*, 2022, pp. 2280–2292.

[142] G. Sebastian, "Privacy and data protection in chatgpt and other ai chatbots: Strategies for securing user information," *International Journal of Security and Privacy in Pervasive Computing (IJSPPC)*, vol. 15, no. 1, pp. 1–14, 2023.

[143] F. D. S. Falcão and E. D. Canedo, "Investigating software development teams members' perceptions of data privacy in the use of large language models (llms)," in *Proceedings of the XXIII Brazilian Symposium on Software Quality*, 2024, pp. 373–382.

[144] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, 2015, pp. 1322–1333.

[145] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *2017 IEEE symposium on security and privacy (SP)*, IEEE, 2017, pp. 3–18.

[146] B. Balle, G. Cherubin, and J. Hayes, "Reconstructing training data with informed adversaries," in *2022 IEEE Symposium on Security and Privacy (SP)*, IEEE, 2022, pp. 1138–1156.

[147] G. Cormode, C. M. Procopiuc, E. Shen, D. Srivastava, and T. Yu, "Empirical privacy and empirical utility of anonymized data," in *2013 IEEE 29th International Conference on Data Engineering Workshops (ICDEW)*, IEEE, 2013, pp. 77–82.

[148] G. Andrew, P. Kairouz, S. Oh, A. Oprea, H. B. McMahan, and V. M. Suriyakumar, "One-shot empirical privacy estimation for federated learning," in *The Twelfth International Conference on Learning Representations*, 2024. [Online]. Available: `https://openreview.net/forum?id=0BqyZSWfzo`.

[149] A. Xiong, T. Wang, N. Li, and S. Jha, "Towards effective differential privacy communication for users' data sharing decision and comprehension," in *2020 IEEE Symposium on Security and Privacy (SP)*, IEEE, 2020, pp. 392–410.

[150] R. Cummings, G. Kaptchuk, and E. M. Redmiles, "" i need a better description": An investigation into user expectations for differential privacy," in *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, 2021, pp. 3037–3052.

[151] P. Nanayakkara, M. A. Smart, R. Cummings, G. Kaptchuk, and E. M. Redmiles, "What are the chances? explaining the epsilon parameter in differential privacy," in *32nd USENIX Security Symposium (USENIX Security 23)*, 2023, pp. 1613–1630.

[152] S. Song, K. Chaudhuri, and A. D. Sarwate, "Stochastic gradient descent with differentially private updates," in *2013 IEEE global conference on signal and information processing*, IEEE, 2013, pp. 245–248.

[153] R. Bassily, A. Smith, and A. Thakurta, "Private empirical risk minimization: Efficient algorithms and tight error bounds," in *IEEE 55th Annual Symposium on Foundations of Computer Science*, FOCS 2014, 2014, pp. 464–473.

[154] M. Nasr et al., "Scalable extraction of training data from (production) language models," *arXiv preprint arXiv:2311.17035*, 2023.

[155] M. Nasr et al., "Scalable extraction of training data from aligned, production language models," in *The Thirteenth International Conference on Learning Representations*, 2025. [Online]. Available: `https://openreview.net/forum?id=vjel3nWP2a`.

[156] C. Zhang, D. Ippolito, K. Lee, M. Jagielski, F. Tramèr, and N. Carlini, "Counterfactual memorization in neural language models," *Advances in Neural Information Processing Systems*, vol. 36, pp. 39 321–39 362, 2023.

[157] A. Schwarzschild, Z. Feng, P. Maini, Z. C. Lipton, and J. Z. Kolter, "Rethinking LLM memorization through the lens of adversarial compression," in *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. [Online]. Available: `https://openreview.net/forum?id=KFmRMvzAZy`.

[158] D. Ippolito et al., "Preventing generation of verbatim memorization in language models gives a false sense of privacy," in *Proceedings of the 16th International Natural Language Generation Conference*, 2023, pp. 28–53.

[159] W. W. Cohen, *Enron email dataset*, Accessed: 2024-12-01, 2004. [Online]. Available: `https://www.cs.cmu.edu/~enron/`.

[160] H. Touvron et al., "Llama 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, 2023.

[161] P. Maini, Z. Feng, A. Schwarzschild, Z. C. Lipton, and J. Z. Kolter, "TOFU: A task of fictitious unlearning for LLMs," in *First Conference on Language Modeling*, 2024. [Online]. Available: `https://openreview.net/forum?id=B41hNBoWLo`.

[162] L. Wutschitz, H. A. Inan, and A. Manoel, *Dp-transformers: Training transformer models with differential privacy*, `https://www.microsoft.com/en-us/research/project/dp-transformers`, Aug. 2022.

[163] E. J. Hu et al., "LoRA: Low-rank adaptation of large language models," in *International Conference on Learning Representations*, 2022. [Online]. Available: `https://openreview.net/forum?id=nZeVKeeFYf9`.

[164] N. Ponomareva et al., "How to dp-fy ml: A practical guide to machine learning with differential privacy," *Journal of Artificial Intelligence Research*, vol. 77, pp. 1113–1201, 2023.

[165] N. Papernot and T. Steinke, "Hyperparameter tuning with renyi differential privacy," in *International Conference on Learning Representations*, 2022. [Online]. Available: `https://openreview.net/forum?id=-70L8lpp9DF`.

[166] M. Nasr, S. Songi, A. Thakurta, N. Papernot, and N. Carlin, "Adversary instantiation: Lower bounds for differentially private machine learning," in *2021 IEEE Symposium on security and privacy (SP)*, IEEE, 2021, pp. 866–882.

[167] M. Nasr et al., "Tight auditing of differentially private machine learning," in *32nd USENIX Security Symposium (USENIX Security 23)*, 2023, pp. 1631–1648.

[168] P. He, "Parameter efficient instruction tuning: An empirical study," *arXiv preprint arXiv:2411.16775*, 2024.

[169] L. Marchyok, N. Carlini, A. Kurakin, and S. Hong, *Evaluating privacy risks of parameter-efficient fine-tuning*, 2025. [Online]. Available: `https://openreview.net/forum?id=i2Ul8WIQm7`.

[170] S. Yeom, I. Giacomelli, M. Fredrikson, and S. Jha, "Privacy risk in machine learning: Analyzing the connection to overfitting," in *2018 IEEE 31st computer security foundations symposium (CSF)*, IEEE, 2018, pp. 268–282.

[171] Y. Ma, X. Zhu, and J. Hsu, "Data poisoning against differentially-private learners: Attacks and defenses," in *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, ser. IJCAI'19, Macao, China, 2019, pp. 4732–4738.

[172] J. Hayes, B. Balle, and S. Mahloujifar, "Bounding training data reconstruction in dp-sgd," *Advances in Neural Information Processing Systems*, vol. 36, 2023.

[173] C. Dwork, A. Roth, et al., "The algorithmic foundations of differential privacy," *Foundations and Trends® in Theoretical Computer Science*, vol. 9, no. 3–4, pp. 211–407, 2014.

[174] R Core Team, *R: A language and environment for statistical computing*, R Foundation for Statistical Computing, Vienna, Austria, 2024. [Online]. Available: `https://www.R-project.org/`.

[175] K. L. van der Veen, R. Seggers, P. Bloem, and G. Patrini, "Three tools for practical differential privacy," in *PPML'18: Privacy Preserving Machine Learning - NeurIPS 2018 Workshop*, 2018. [Online]. Available: `https://arxiv.org/abs/1812.02890`.

[176] A. Kurakin, S. Song, S. Chien, R. Geambasu, A. Terzis, and A. Thakurta, "Toward training at imagenet scale with differential privacy," *arXiv preprint arXiv:2201.12328*, 2022.

[177] Z. Ren, Y. J. Lee, and M. S. Ryoo, "Learning to anonymize faces for privacy preserving action detection," in *Proceedings of the european conference on computer vision (ECCV)*, 2018, pp. 620–636.

[178] A. Yale, S. Dash, R. Dutta, I. Guyon, A. Pavao, and K. P. Bennett, "Privacy preserving synthetic health data," in *ESANN 2019-European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, 2019.

[179] B. Kulynych, J. F. Gomez, G. Kaissis, F. Calmon, and C. Troncoso, "Attack-aware noise calibration for differential privacy," in *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. [Online]. Available: `https://openreview.net/forum?id=hOcsUrOYOD`.

[180] G. Kaissis, S. Kolek, B. Balle, J. Hayes, and D. Rueckert, "Beyond the calibration point: Mechanism comparison in differential privacy," in *International Conference on Machine Learning*, 2024.

[181] A. Hard et al., "Federated learning for mobile keyboard prediction," *arXiv preprint arXiv:1811.03604*, 2018.

[182] Y. Zhang, Z. Xu, S. Wu, Y. Zhang, and D. Ramage, "Synthesizing and adapting error correction data for mobile large language model applications," in *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track)*, G. Rehm and Y. Li, Eds., Vienna, Austria: Association for Computational Linguistics, Jul. 2025, pp. 1102–1112, ISBN: 979-8-89176-288-6. DOI: `10.18653/v1/2025.acl-industry.78`. [Online]. Available: `https://aclanthology.org/2025.acl-industry.78/`.

[183] A. Karatzoglou and B. Hidasi, "Deep learning for recommender systems," in *Proceedings of the eleventh ACM conference on recommender systems*, 2017, pp. 396–397.

[184] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep learning based recommender system: A survey and new perspectives," *ACM computing surveys (CSUR)*, vol. 52, no. 1, pp. 1–38, 2019.

[185] X. Yue et al., "Synthetic text generation with differential privacy: A simple and practical recipe," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2023, pp. 1321–1342.

[186] A. Kurakin, N. Ponomareva, U. Syed, L. MacDermed, and A. Terzis, "Harnessing large-language models to generate private synthetic text," *arXiv preprint arXiv:2306.01684*, 2023.

[187] C. Xie et al., "Differentially private synthetic data via foundation model APIs 2: Text," in *Forty-first International Conference on Machine Learning*, 2024. [Online]. Available: `https://openreview.net/forum?id=LWD7upg1ob`.

[188] C. Hou et al., "Pre-text: Training language models on private federated data in the age of LLMs," in *Forty-first International Conference on Machine Learning*, 2024. [Online]. Available: `https://openreview.net/forum?id=3WCvnkHnxV`.

[189] D. Yu, P. Kairouz, S. Oh, and Z. Xu, "Privacy-preserving instructions for aligning large language models," in *Forty-first International Conference on Machine Learning*, 2024. [Online]. Available: `https://openreview.net/forum?id=mUT1biz09t`.

[190] C. Hou, M.-Y. Wang, Y. Zhu, D. Lazar, and G. Fanti, "Private federated learning using preference-optimized synthetic data," in *Forty-second International Conference on Machine Learning*, 2025. [Online]. Available: `https://openreview.net/forum?id=ZuaU2bYzlc`.

[191] B. Tan, Z. Xu, E. P. Xing, Z. Hu, and S. Wu, "Synthesizing privacy-preserving text data via finetuning *without* finetuning billion-scale LLMs," in *Forty-second International Conference on Machine Learning*, 2025. [Online]. Available: https://openreview.net/forum?id=FCm4laCLiH.

[192] N. S. Keskar, B. McCann, L. R. Varshney, C. Xiong, and R. Socher, "Ctrl: A conditional transformer language model for controllable generation," *arXiv preprint arXiv:1909.05858*, 2019.

[193] S. Dathathri et al., "Plug and play language models: A simple approach to controlled text generation," in *International Conference on Learning Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=H1edEyBKDS.

[194] M. Przystupa and M. Abdul-Mageed, "Neural machine translation of low-resource and similar languages with backtranslation," in *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, 2019, pp. 224–235.

[195] A.-S. Charest, "How can we analyze differentially-private synthetic datasets?" *Journal of Privacy and Confidentiality*, vol. 2, no. 2, 2011.

[196] J. Near and D. Darais, *Differentially private synthetic data*, NIST Cybersecurity Insights Blog, Accessed: 2025-09-12, May 2021. [Online]. Available: https://www.nist.gov/blogs/cybersecurity-insights/differentially-private-synthetic-data.

[197] Z. Lin, S. Gopi, J. Kulkarni, H. Nori, and S. Yekhanin, "Differentially private synthetic data via foundation model APIs 1: Images," in *The Twelfth International Conference on Learning Representations*, 2024. [Online]. Available: https://openreview.net/forum?id=YEhQs8POIo.

[198] R. McKenna, B. Mullins, D. Sheldon, and G. Miklau, "Aim: An adaptive and iterative mechanism for differentially private synthetic data," *Proceedings of the VLDB Endowment*, vol. 15, no. 11, pp. 2599–2612, 2022.

[199] Y. Tao, R. McKenna, M. Hay, A. Machanavajjhala, and G. Miklau, "Benchmarking differentially private synthetic data generation algorithms," *arXiv preprint arXiv:2112.09238*, 2021.

[200] K. Chen, X. Li, C. Gong, R. McKenna, and T. Wang, "Benchmarking differentially private tabular data synthesis," *arXiv preprint arXiv:2504.14061*, 2025.

[201] Z. Zhao, Q. Jin, F. Chen, T. Peng, and S. Yu, "A large-scale dataset of patient summaries for retrieval-based clinical decision support systems.," *Scientific data*, vol. 10 1, p. 909, 2023. [Online]. Available: `https://api.semanticscholar.org/CorpusID:266360591`.

[202] Yelp, Inc., *Yelp open dataset*, Accessed: 2025-09-12, 2025. [Online]. Available: `https://business.yelp.com/data/resources/open-dataset/`.

[203] G. Team et al., "Gemma 3 technical report," *arXiv preprint arXiv:2503.19786*, 2025.

[204] A. Y. Qwen et al., "Qwen2. 5 technical report," *arXiv preprint*, 2024.

[205] K. Pillutla et al., "Mauve: Measuring the gap between neural text and human text using divergence frontiers," *Advances in Neural Information Processing Systems*, vol. 34, pp. 4816–4828, 2021.

[206] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[207] L. Gao, J. Schulman, and J. Hilton, "Scaling laws for reward model overoptimization," in *International Conference on Machine Learning*, PMLR, 2023, pp. 10 835–10 866.

[208] J. Eisenstein et al., "Helping or herding? reward model ensembles mitigate but do not eliminate reward hacking," in *First Conference on Language Modeling*, 2024. [Online]. Available: `https://openreview.net/forum?id=5u1GpUkKtG`.

[209] J. Lin, "Divergence measures based on the shannon entropy," *IEEE Transactions on Information theory*, vol. 37, no. 1, pp. 145–151, 2002.

[210] R. Bommasani, S. Wu, and X. Schofield, "Towards private synthetic text generation," in *NeurIPS 2019 Machine Learning with Guarantees Workshop*, 2019.

[211] J. Mattern, Z. Jin, B. Weggenmann, B. Schoelkopf, and M. Sachan, "Differentially private language models for secure data sharing," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Y. Goldberg, Z. Kozareva, and Y. Zhang, Eds., Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 4860–4873. DOI: 10.18653/v1/2022.emnlp-main.323. [Online]. Available: https://aclanthology.org/2022.emnlp-main.323/.

[212] A. Carranza, R. Farahani, N. Ponomareva, A. Kurakin, M. Jagielski, and M. Nasr, "Synthetic query generation for privacy-preserving deep retrieval systems using differentially private language models," in *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, K. Duh, H. Gomez, and S. Bethard, Eds., Mexico City, Mexico: Association for Computational Linguistics, Jun. 2024, pp. 3920–3930. DOI: 10.18653/v1/2024.naacl-long.217. [Online]. Available: https://aclanthology.org/2024.naacl-long.217/.

[213] S. Ochs and I. Habernal, "Private synthetic text generation with diffusion models," in *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, L. Chiruzzo, A. Ritter, and L. Wang, Eds., Albuquerque, New Mexico: Association for Computational Linguistics, Apr. 2025, pp. 10 612–10 626, ISBN: 979-8-89176-189-6. DOI: 10.18653/v1/2025.naacl-long.532. [Online]. Available: https://aclanthology.org/2025.naacl-long.532/.

[214] P. Putta, A. Steele, and J. W. Ferrara, *Differentially private conditional text generation for synthetic data production*, 2023. [Online]. Available: https://openreview.net/forum?id=LUql3ZOFwFD.

[215] G. DeSalvo, J.-F. Kagy, L. Karydas, A. Rostamizadeh, and S. Kumar, "Softsrv: Learn to generate targeted synthetic data," *arXiv preprint arXiv:2410.16534*, 2024.

[216] DeepSeek-AI, *Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning*, 2025. arXiv: `2501.12948 [cs.CL]`. [Online]. Available: `https://arxiv.org/abs/2501.12948`.

[217] Q. Yu et al., "Dapo: An open-source llm reinforcement learning system at scale," *arXiv preprint arXiv:2503.14476*, 2025.

[218] X. Li, H. Zou, and P. Liu, "Limr: Less is more for rl scaling," *arXiv preprint arXiv:2502.11886*, 2025.

[219] T. Shi, Y. Wu, L. Song, T. Zhou, and J. Zhao, "Efficient reinforcement finetuning via adaptive curriculum learning," *arXiv preprint arXiv:2504.05520*, 2025.

[220] Y. E. Xu, Y. Savani, F. Fang, and Z. Kolter, "Not all rollouts are useful: Down-sampling rollouts in llm reinforcement learning," *arXiv preprint arXiv:2504.13818*, 2025.

[221] Y. Wang et al., "Reinforcement learning for reasoning in large language models with one training example," *arXiv preprint arxiv:2504.20571*, 2025.

[222] L. Bereska and S. Gavves, "Mechanistic interpretability for AI safety - a review," *Transactions on Machine Learning Research*, 2024, Survey Certification, Expert Certification, ISSN: 2835-8856. [Online]. Available: `https://openreview.net/forum?id=ePUVetPKu6`.

[223] J. Dong, A. Roth, and W. J. Su, "Gaussian differential privacy," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 84, no. 1, pp. 3–37, 2022.

[224] M. B. Cohen, J. Nelson, and D. P. Woodruff, "Optimal approximate matrix product in terms of stable rank," in *43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016)*, Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2016, pp. 11–1.

[225] S. Dirksen, "Tail bounds via generic chaining," *Electronic Journal of Probability*, vol. 20, no. 53, pp. 1–29, 2015.

[226] R. E. Paley and A. Zygmund, "On some series of functions,(3)," in *Mathematical Proceedings of the Cambridge Philosophical Society*, Cambridge University Press, vol. 28, 1932, pp. 190–205.

[227] M. Towers et al., *Gymnasium: A standard interface for reinforcement learning environments*, 2024. arXiv: `2407.17032 [cs.LG]`. [Online]. Available: `https://arxiv.org/abs/2407.17032`.

[228] M. Chevalier-Boisvert et al., "Minigrid & miniworld: Modular & customizable reinforcement learning environments for goal-oriented tasks," *CoRR*, vol. abs/2306.13831, 2023.

[229] E. Leurent, *An environment for autonomous driving decision-making*, `https://github.com/eleurent/highway-env`, 2018.

[230] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, "Stable-baselines3: Reliable reinforcement learning implementations," *Journal of Machine Learning Research*, vol. 22, no. 268, pp. 1–8, 2021. [Online]. Available: `http://jmlr.org/papers/v22/20-1364.html`.

[231] L. von Werra et al., *Trl: Transformer reinforcement learning*, `https://github.com/huggingface/trl`, 2020.

[232] S. Gehman, S. Gururangan, M. Sap, Y. Choi, and N. A. Smith, "Realtoxicityprompts: Evaluating neural toxic degeneration in language models," *arXiv preprint arXiv:2009.11462*, 2020.

[233] B. Vidgen, T. Thrush, Z. Waseem, and D. Kiela, "Learning from the worst: Dynamically generated datasets to improve online hate detection," in *ACL*, 2021.

[234] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, "High-dimensional continuous control using generalized advantage estimation," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2016.

[235] S. Bae, J. Hong, M. Y. Lee, H. Kim, J. Nam, and D. Kwak, "Online difficulty filtering for reasoning oriented reinforcement learning," *arXiv preprint arXiv:2504.03380*, 2025.

[236] A. Gokaslan, V. Cohen, E. Pavlick, and S. Tellex, *Openwebtext corpus*, `http://Skylion007.github.io/OpenWebTextCorpus`, 2019.

[237] L. Chua et al., "Mind the privacy unit! user-level differential privacy for language model fine-tuning," in *First Conference on Language Modeling*, 2024. [Online]. Available: `https://openreview.net/forum?id=Jd0bCD12DS`.

[238] Z. Charles et al., "Fine-tuning large language models with user-level differential privacy," in *ICML 2024 Workshop on Theoretical Foundations of Foundation Models*, 2024. [Online]. Available: `https://openreview.net/forum?id=Pb7TFHQtjw`.

[239] A. Zou, Z. Wang, N. Carlini, M. Nasr, J. Z. Kolter, and M. Fredrikson, "Universal and transferable adversarial attacks on aligned language models," *arXiv preprint arXiv:2307.15043*, 2023.

[240] H. Duan, A. Dziedzic, M. Yaghini, N. Papernot, and F. Boenisch, "On the privacy risk of in-context learning," in *The 61st Annual Meeting Of The Association For Computational Linguistics*, 2023.

[241] H. Duan, A. Dziedzic, N. Papernot, and F. Boenisch, "Flocks of stochastic parrots: Differentially private prompt learning for large language models," *Advances in Neural Information Processing Systems*, 2023.

[242] T. Wu, A. Panda, J. T. Wang, and P. Mittal, "Privacy-preserving in-context learning for large language models," in *International Conference on Learning Representations*, 2024.

[243] X. Tang et al., "Privacy-preserving in-context learning with differentially private few-shot generation," *International Conference on Learning Representations*, 2024.

[244] J. Hong, J. T. Wang, C. Zhang, Z. Li, B. Li, and Z. Wang, "Dp-opt: Make large language model your privacy-preserving prompt engineer," *International Conference on Learning Representations*, 2024.

[245] I. Mironov, "Rényi differential privacy," in *2017 IEEE 30th computer security foundations symposium (CSF)*, IEEE, 2017, pp. 263–275.

[246] Y.-X. Wang, B. Balle, and S. P. Kasiviswanathan, "Subsampled rényi differential privacy and analytical moments accountant," in *The 22nd international conference on artificial intelligence and statistics*, PMLR, 2019, pp. 1226–1235.

[247] L. von Werra et al., *TRL: Transformer Reinforcement Learning*, version 0.25. [Online]. Available: `https://github.com/huggingface/trl`.

[248] A. Holtzman, J. Buys, L. Du, M. Forbes, and Y. Choi, "The curious case of neural text degeneration," in *International Conference on Learning Representations*, 2020. [Online]. Available: `https://openreview.net/forum?id=rygGQyrFvH`.

[249] A. Singh, M. D'Arcy, A. Cohan, D. Downey, and S. Feldman, "Scirepeval: A multi-format benchmark for scientific document representations," in *Conference on Empirical Methods in Natural Language Processing*, 2022. [Online]. Available: `https://api.semanticscholar.org/CorpusID:254018137`.

[250] Y. Gu et al., *Domain-specific language model pretraining for biomedical natural language processing*, 2020. eprint: `arXiv:2007.15779`.

[251] I. Beltagy, K. Lo, and A. Cohan, "Scibert: A pretrained language model for scientific text," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 3615–3620.

[252] P. Singhal, T. Goyal, J. Xu, and G. Durrett, "A long way to go: Investigating length correlations in RLHF," in *First Conference on Language Modeling*, 2024. [Online]. Available: `https://openreview.net/forum?id=G8LaO1P0xv`.

[253] Y. Zhang et al., "Qwen3 embedding: Advancing text embedding and reranking through foundation models," *arXiv preprint arXiv:2506.05176*, 2025.

[254] X. Wu, T. Nguyen, D. Zhang, W. Y. Wang, and A. T. Luu, "Fastopic: Pretrained transformer is a fast, adaptive, stable, and transferable topic model," *Advances in Neural Information Processing Systems*, vol. 37, pp. 84 447–84 481, 2024.