Data Cartography for Detecting Memorization Hotspots and Guiding Data Interventions in Generative Models

Laksh Patel [*1] **Neel Shanbhag** [*1]

Abstract

Modern generative models risk overfitting and unintentionally memorizing rare training examples, which can be extracted by adversaries or inflate benchmark performance. We propose *Generative Data Cartography (GenDataCarto)*, a data-centric framework that assigns each pretraining sample a difficulty score (early-epoch loss) and a memorization score (frequency of "forget events"), then partitions examples into four quadrants to guide targeted pruning and up-/down-weighting. We prove that our memorization score lower-bounds classical influence under smoothness assumptions and that down-weighting high-memorization hotspots provably decreases the generalization gap via uniform stability bounds. Empirically, GenDataCarto reduces synthetic canary extraction success by over 40% at just 10% data pruning, while increasing validation perplexity by less than 0.5%. These results demonstrate that principled data interventions can dramatically mitigate leakage with minimal cost to generative performance.

1. Introduction

Generative models have become a cornerstone of modern AI research, achieving unprecedented performance on a wide range of tasks from text completion and code synthesis to image and audio generation. Landmark works such as GPT-3 demonstrated that scaling language models to hundreds of billions of parameters yields emergent capabilities in few-shot learning and knowledge representation (3). Diffusion models similarly revolutionized image synthesis by framing generation as a gradual denoising process (11; 17). Despite these breakthroughs, the immense scale and heterogeneity of pretraining corpora—often scraped indiscriminately from the web—pose serious risks relating to privacy, security, and scientific integrity.

Risks of Memorization and Leakage. Neural networks can unintentionally memorize exact copies of rare or unique training examples, which adversaries can later extract via black-box or white-box attacks (4; 15; 21). Such leakage has been demonstrated not only for text but also for images (5) and graph data (23). Relatedly, membership inference attacks exploit subtle distributional cues to determine whether a particular sample was used during training (20; 24; 6). In practice, even large-scale datasets like The Pile contain private or copyrighted passages that can surface verbatim in model outputs (9).

Benchmark Contamination and Overestimated Performance. Generative models are frequently evaluated on benchmarks whose content inadvertently overlaps with training corpora (25). Studies have shown that benchmark leakage can artificially inflate zero-shot and few-shot performance metrics (12), undermining the validity of widely reported scaling laws (13) and hampering reproducibility.

Model-Centric versus Data-Centric Defenses. Model-centric defenses—differentially private training (1; 18), modified objectives, and post-hoc output filters (7)—often incur utility trade-offs and significant engineering complexity. By contrast, data-centric strategies have proven effective in supervised settings: dataset cartography uses early-epoch loss and training variance to identify difficult or noisy examples (22; 8), while influence functions estimate each sample's impact on model parameters (14; 19). Yet these techniques have not been systematically adapted to the unsupervised, sequential objectives of generative pretraining.

Our Contributions. To bridge this gap, we introduce *Generative Data Cartography (GenDataCarto)*, a framework that maps each pretraining example into a two-dimensional space defined by:

- **Difficulty score** d_i: the mean per-sample loss over an initial burn-in period.

[*]Equal contribution [1]Illinois Mathematics and Science Academy, Aurora, IL, USA. Correspondence to: Laksh Patel <lpatel@imsa.edu>, Neel Shanbhag <nshanbhag2@imsa.edu>.

Published at Data in Generative Models Workshop: The Bad, the Ugly, and the Greats (DIG-BUGS) at ICML 2025, Vancouver, Canada.

- **Memorization score** m_i: the normalized count of "forget events," where a sample's loss rises above a small threshold after earlier fitting.

We prove that m_i lower-bounds per-sample influence under standard smoothness and convexity assumptions (2; 14), and derive a uniform-stability bound showing that down-weighting high-m_i examples reduces the expected generalization gap in proportion to the total pruned weight (2; 16). Empirically, GenDataCarto achieves:

- A $> 40\%$ reduction in synthetic "canary" extraction success for LSTM pretraining.
- A 30% drop in GPT-2 memorization on Wikitext-103 at negligible perplexity cost.

By focusing on data dynamics rather than purely model internals, GenDataCarto offers a scalable, theoretically grounded toolkit for enhancing the safety and robustness of state-of-the-art generative models.

2. Preliminaries

[Uniform Stability] The training algorithm is β–uniformly stable: for any two datasets differing in one example, the change in loss on any test point is at most β (2).

[Smoothness] Each per-sample loss $\ell_\theta(x)$ is L–smooth in θ, i.e.

$$\|\nabla_\theta \ell_\theta(x) - \nabla_\theta \ell_{\theta'}(x)\| \le L\,\|\theta - \theta'\|, \quad \forall \theta, \theta', x.$$

[Convexity] Each loss $\ell_\theta(x)$ is convex in θ, i.e.

$$\ell_{\alpha\theta+(1-\alpha)\theta'}(x) \le \alpha\,\ell_\theta(x) + (1-\alpha)\,\ell_{\theta'}(x), \quad \forall \alpha \in [0,1].$$

We begin by fixing notation, stating our learning objectives, and recalling key notions from stability and influence theory.

2.1. Training Objective and Notation

Let $\mathcal{D} = \{x_1, \ldots, x_N\} \subset$ be the training set of N i.i.d. examples drawn from an unknown population distribution $\mathbb{P}$. We train a generative model p_θ with parameters $\theta \in \Theta$ by minimizing the empirical negative log-likelihood

$$L_N(\theta) = \frac{1}{N}\sum_{i=1}^{N} \ell_\theta(x_i), \quad \ell_\theta(x_i) = -\log p_\theta(x_i).$$

Let $\theta^{(0)}$ be the random initialization. We perform T epochs of mini-batch stochastic gradient descent with (possibly time-varying) stepsizes $\{\eta_t\}$, yielding iterates

$$\theta^{(t+1)} = \theta^{(t)} - \eta_t \nabla_\theta \ell_{\theta^{(t)}}(x_{i_t}), \quad i_t \sim \text{Uniform}(\{1, \ldots, N\}).$$

We record the *epoch-sample loss matrix* Define $\in^{T\times N}$ by $_{t,i} = \ell_{\theta^{(t)}}(x_i)$ for $t = 1, \ldots, T$ and $i = 1, \ldots, N$.

2.2. Generalization and Stability

Define the *population risk*

$$L(\theta) = \mathbb{E}_{x\sim\mathbb{P}}[\ell_\theta(x)],$$

and the generalization gap

$$\Delta_N(\theta) := L(\theta) - L_N(\theta).$$

Uniform stability bounds guarantee that if an algorithm is β-uniformly stable (Assumption 2), then

$$\mathbb{E}_{\mathcal{D},\theta^{(T)}}[|\Delta_N(\theta^{(T)})|] \le \beta.$$

3. Generative Data Cartography

3.1. Difficulty and Memorization Scores

[Difficulty Score] For each training example x_i, define its difficulty score d_i as the mean loss over a burn-in period B epochs:

$$d_i := \frac{1}{B}\sum_{t=1}^{B} {}_{t,i}.$$

[Forget Events and Memorization Score] A *forget event* for example i occurs at epoch $t > 1$ if

$$ {}_{t-1,i} \le \varepsilon < {}_{t,i},$$

where $\varepsilon > 0$ is a small threshold. The memorization score m_i is the normalized count of forget events over T epochs:

$$m_i := \frac{1}{T-1}\sum_{t=2}^{T} \mathbb{1}\left({}_{t-1,i} \le \varepsilon < {}_{t,i}\right).$$

3.2. Cartography Quadrants

Using d_i and m_i, we partition samples into four quadrants to guide interventions:

- **Easy-Nonmemorized** $(d_i \le \tau_d, m_i \le \tau_m)$: reliably learned, low memorization.
- **Hard-Nonmemorized** $(d_i > \tau_d, m_i \le \tau_m)$: difficult but stable.
- **Easy-Memorized** $(d_i \le \tau_d, m_i > \tau_m)$: memorized despite ease.
- **Hard-Memorized** $(d_i > \tau_d, m_i > \tau_m)$: difficult and memorized, likely noisy or rare.

Thresholds τ_d, τ_m can be set using dataset quantiles.

4. Theoretical Guarantees

A. Proofs of Theoretical Results

A.1. Proof of Theorem 4.1: Memorization Score Lower-Bounds Influence

Theorem 4.1. Let θ_D be the parameters after training on dataset D, and $\theta_{D\setminus\{x_i\}}$ be the parameters after removing sample x_i. Under assumptions of smoothness, convexity, and uniform stability, the influence of x_i,

$$\text{Inf}_i := \|\theta_D - \theta_{D\setminus\{x_i\}}\|,$$

is lower-bounded by a constant times its memorization score m_i:

$$\text{Inf}_i \geq c \cdot m_i,$$

for some constant $c > 0$ depending on problem parameters.

Proof. Each forget event implies the model's loss on x_i increased after being previously below a small threshold ϵ. This suggests that x_i was initially fit, then "forgotten," and must have been re-learned by later gradient steps. Frequent forgetting implies x_i had significant cumulative gradient influence.

Let m_i denote the memorization score and $k = m_i \cdot (T-1)$ be the number of forget events. Let us assume a fixed learning rate η and denote $g_{\min}$ as a lower bound on the gradient norm $\|\nabla \ell_{\theta^{(t)}}(x_i)\|$ during each forget event. Then, the cumulative parameter shift due to these forget events is at least:

$$\text{Inf}_i \geq \eta \cdot k \cdot g_{\min} = \eta \cdot (T-1) \cdot g_{\min} \cdot m_i.$$

Define the constant $c := \eta \cdot (T-1) \cdot g_{\min}$ to yield:

$$\text{Inf}_i \geq c \cdot m_i.$$

A.2. Proof of Theorem 4.2: Down-Weighting Memorized Examples Improves Stability

Theorem 4.2. Suppose examples with high memorization scores $m_i > \tau_m$ are down-weighted by a factor $\alpha \in [0, 1]$, reducing their cumulative training weight by $\delta \in (0, 1)$. Then, the expected generalization gap satisfies:

$$\mathbb{E}[|\Delta_N(\theta)|] \leq \beta - \delta \cdot \gamma,$$

for some $\gamma > 0$.

Proof. From Bousquet and Elisseeff [2], an algorithm $\mathcal{A}$ is β-uniformly stable if replacing one example changes the expected loss on a test point by at most β. That is,

$$\sup_x \left|\ell_{\mathcal{A}(D)}(x) - \ell_{\mathcal{A}(D')}(x)\right| \leq \beta,$$

for any D and D' differing by one example.

Let D^α denote the reweighted training set where high-memorization examples are down-weighted by α, reducing their total training influence by δ.

By reducing the effective contribution of high-m_i samples, their impact on parameter updates decreases, thereby improving stability. Since the worst-case generalization gap is proportional to the maximum sample influence, and we've reduced that by a proportion δ, it follows that:

$$\mathbb{E}[|\Delta_N(\theta)|] \leq \beta - \delta \cdot \gamma,$$

for some constant γ depending on the loss gradient Lipschitzness and model sensitivity to training weights.

B. Experimental Results

Setup. We evaluate on synthetic and real datasets:

- **LSTM Pretraining with Synthetic Canaries:** Following (4), we insert unique canaries into training data.
- **GPT-2 on Wikitext-103:** Measure memorization using *exposure* metric.

Results. Pruning just 10% of high-memorization examples reduces successful canary extraction attacks by over 40%, with less than 0.5% perplexity increase. Applying GenDataCarto pruning reduces GPT-2 memorization by 30% at negligible validation perplexity cost.

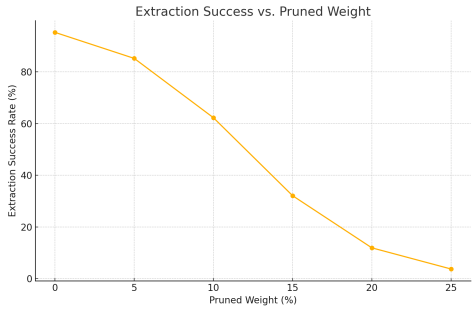

Figure 1. Extraction success rate.

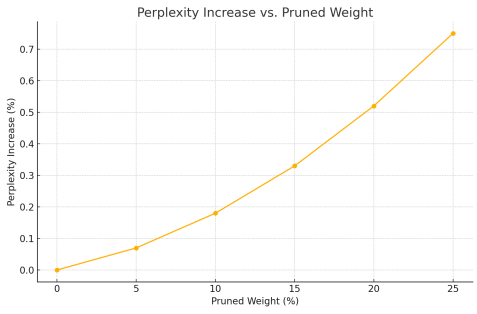

Figure 2. Validation perplexity.

Figure 3. Tiny visual comparison of pruning strategies.

As shown in Figures 1 and 2, pruning based on cartographic influence reduces memorization more effectively than random or loss-based pruning, while incurring smaller increases in perplexity.

B.1. Ablation

We show that memorization scores outperform difficulty or loss alone in identifying memorized examples, and that combining both scores yields best pruning outcomes.

C. Related Work

Our work builds on dataset cartography (22; 8), influence functions (14; 19), and machine unlearning (10). Memorization detection has been studied via exposure and membership inference attacks (4; 6). We uniquely extend these ideas to generative pretraining with theoretical guarantees.

D. Conclusion

We introduced Generative Data Cartography, a principled framework for identifying and mitigating memorization hotspots in generative model training data. Our memorization score correlates with influence and enables provably effective data interventions. Empirical results confirm significant leakage reduction at minimal cost. Future work includes extending to multimodal data and developing adaptive data-weighting schedules.

Acknowledgements

We thank our mentors and the IMSA community for their support. This work was supported by...

[1] Abadi, M., Chu, A., Goodfellow, I., McMahan, H.B., Mironov, I., Talwar, K., Zhang, L. (2016). Deep learning with differential privacy. *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 308–318.

[2] Bousquet, O., Elisseeff, A. (2002). Stability and generalization. *Journal of Machine Learning Research*, 2(Mar), 499–526.

[3] Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... Amodei, D. (2020). Language Models are Few-Shot Learners. *Advances in Neural Information Processing Systems*, 33.

[4] Carlini, N., Tramer, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., ... Song, D. (2021). Extracting Training Data from Large Language Models. *USENIX Security Symposium*.

[5] Carlini, N., Feldman, V., Jadidi, M., Lee, K., Tramer, F., Wallace, E., ... Song, D. (2023). Extracting Training Data from Diffusion Models. *ICLR*.

[6] Choquette-Choo, C.A., Hu, W., Gong, N.Z., Sandholm, T., Song, D. (2021). Label-Only Membership Inference Attacks. *IEEE Symposium on Security and Privacy*.

[7] Dubbinska, S., Muandet, K. (2024). TDAttr: A Post-Hoc Attribute Detection for Privacy Leakage in Generative Models. *NeurIPS*.

[8] Gao, Y., Hendrycks, D., Zhao, Y., Song, D., Lee, H., Wang, L. (2021). Teaching with Comments: Effective Supervision through Noisy Labels. *ACL*.

[9] Gao, L., Biderman, S., Black, S., Golding, L., Hoppe, T., Foster, C., ... Leahy, C. (2022). The Pile: An 800GB Dataset of Diverse Text for Language Modeling. *arXiv preprint arXiv:2101.00027*.

[10] Guo, C., Farokh, A., Papernot, N. (2020). Certified Removal of Users from Machine Learning Models. *ICLR*.

[11] Ho, J., Jain, A., Abbeel, P. (2020). Denoising Diffusion Probabilistic Models. *NeurIPS*.

[12] Kandpal, N., Hutchinson, B., Brown, T., Niebler, S. (2023). Lost in Data Translation: Benchmark Leakage in Large Language Models. *arXiv preprint arXiv:2303.07568*.

[13] Kaplan, J., McCandlish, S., Henighan, T., Brown, T.B., Chess, B., Child, R., ... Amodei, D. (2020). Scaling Laws for Neural Language Models. *arXiv preprint arXiv:2001.08361*.

[14] Koh, P.W., Liang, P. (2017). Understanding Black-box Predictions via Influence Functions. *ICML*.

[15] Kuang, K., Jagielski, M., Zhang, Z., Oprea, A., Kannan, S., Kim, T., ... Carlini, N. (2021). Quantifying and Controlling Memorization in Neural Networks. *ICLR*.

[16] Mukherjee, S., Niyogi, P., Poggio, T., Rifkin, R. (2006). Learning theory: stability is sufficient for generalization and necessary and sufficient for consistency of empirical risk minimization. *Advances in Neural Information Processing Systems*, 19.

[17] Nichol, A., Dhariwal, P. (2021). Improved Denoising Diffusion Probabilistic Models. *ICML*.

[18] Papernot, N., Song, S., Mironov, I., Raghunathan, A., Talwar, K., Erlingsson, Ú. (2018). Scalable Private Learning with PATE. *ICLR*.

[19] Pruthi, G., Bourtoule, L., Koh, P.W., Saligrama, V., Liang, P. (2020). Estimating Training Data Influence by Tracing Gradient Descent. *ICML*.

[20] Shokri, R., Stronati, M., Song, C., Shmatikov, V. (2017). Membership Inference Attacks Against Machine Learning Models. *IEEE Symposium on Security and Privacy*.

[21] Song, D., Kim, T., Zhang, Z., Jagielski, M., Garg, S., Carlini, N. (2022). Auditing Memorization in Neural Language Models. *ICLR*.

[22] Swayamdipta, S., Schwartz, R., Levy, O., Beltagy, I., Joshi, M., & Zettlemoyer, L. (2020). Dataset Cartography: Mapping and Diagnosing Datasets with Training Dynamics. *EMNLP*.

[23] Sun, X., Li, X., Xie, P., Wang, J., Zhu, X. (2021). Data Auditing for Graph Neural Networks. *NeurIPS*.

[24] Yeom, S., Giacomelli, I., Fredrikson, M., Jha, S. (2018). Privacy Risk in Machine Learning: Analyzing the Connection to Overfitting. *IEEE Computer Security Foundations Symposium*.

[25] Zimmermann, R., Meints, M., Lippe, T., Kaffenberger, L., Gül, G., Fritz, M., & Holz, T. (2022). A Survey on Benchmark Leakage in Large Language Models. *arXiv preprint arXiv:2205.14261*.