Layer-wise Influence Tracing: Data-Centric Mitigation of Memorization in Diffusion Models

Thomas Y. Chen [1]

Abstract

Text-to-image diffusion models can inadvertently memorize and regenerate unique training images, posing serious privacy and copyright risks. While recent work links such memorization to sharp spikes in the model's log-density Hessian, existing diagnostics stop at flagging *that* a model overfits, not *which* samples are to blame or how to remove them. We introduce *layer-wise influence tracing*, a scalable Hessian decomposition that assigns every training image a curvature-based influence score. Deleting only the top $1\,\%$ high-risk images and performing a single, low-learning-rate fine-tune cuts verbatim reconstructions in Stable Diffusion XL by 72% while keeping Fréchet Inception Distance within 1% of the baseline. The full procedure costs just 2.3 GPU-hours—over an order of magnitude cheaper than full-Hessian methods—and yields similar gains on a 1-billion-parameter distilled backbone. Our results turn a coarse memorization signal into an actionable, data-centric mitigation strategy, paving the way toward privacy-respecting generative models at 10 B+ scale.

1. Introduction

Text-to-image diffusion models such as Stable Diffusion and DALL·E have crossed from research demos to everyday creative and commercial tools. Yet a mounting body of evidence shows that these models can "verbatim-copy" rare or unique training images, posing privacy, intellectual-property, and ethical hazards. Lawsuits brought by artists and photographers, and empirical analyses that retrieve copyrighted material with near-pixel accuracy, illustrate the gravity of the problem (Somepalli et al., 2023).

Most current safety audits look for *global* signs of overfitting—e.g. counting near-duplicate generations or examining curvature statistics of the learned density. A recent study finds that sharp, low-dimensional pockets in the probability landscape, revealed by large-magnitude eigenvalues of the log-density Hessian, correlate strongly with memorization events in latent diffusion models (Jeon et al., 2024). While this insight provides a powerful diagnostic, it remains a coarse-grained signal: it tells us *that* the model has memorized, but not *which specific training images* are responsible, nor how to surgically remove them without harming overall image quality.

We tackle this gap with a *layer-wise influence tracing* framework. By factorizing the Hessian through layer Jacobians and combining it with influence-function theory, we compute per-sample "risk scores" that predict how much deleting a single image would flatten the surrounding landscape—and therefore reduce copy-risk—at negligible computational overhead. A targeted, single-epoch "surgical" fine-tune on Stable Diffusion XL shows that excising the top $1\,\%$ riskiest images lowers verbatim reconstructions by 70% while keeping the Fréchet Inception Distance (FID) within 1% of baseline.

Our contributions are threefold: **(i)** a scalable estimator that decomposes curvature-based memorization signals down to individual training samples; **(ii)** an efficient stochastic Lanczos routine that makes the method practical for billion-parameter diffusion backbones; and **(iii)** the first demonstration that influence-guided, data-centric remediation can meaningfully cut memorization without aesthetic degradation, advancing the workshop's agenda of trustworthy generative AI through responsible data management.

2. Related Work

Memorization in generative models. Early work on unintended memorization exposed how language models reproduce rare training sequences (Carlini et al., 2019). For diffusion models, Somepalli et al. (2023) and Carlini et al. (2023) demonstrate near-pixel reconstructions of copyrighted or private images, underscoring privacy risks and sparking litigation from affected creators.

[1]Department of Computer Science, Fu Foundation School of Engineering and Applied Science, Columbia University, New York, NY 10027, USA. Correspondence to: Thomas Y. Chen <chen.thomas@columbia.edu>.

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

Curvature-based diagnostics. Sharp "spikes" in a model's probability landscape correlate with overfitting: Jeon et al. (2024) relate large log-density Hessian eigenvalues to copy risk, while Jiang et al. (2020) extend the connection to gradient-variance measures. These methods, however, yield only aggregate scores and lack sample-level guidance.

Sample-level attribution. Influence functions translate second-order information into per-example impact estimates (Koh & Liang, 2017). Extensions to deep generative settings remain sparse; Basu et al. (2021) adapt the idea to variational auto-encoders but require expensive retraining for each deletion, limiting practicality on billion-parameter diffusion backbones.

Data-centric mitigation. Research on machine unlearning and targeted data deletion shows that removing a small subset of influential examples can curb overfitting while preserving accuracy (Ginart et al., 2019). Parallel efforts in data-centric AI advocate systematic dataset hygiene to improve robustness and fairness (Zha et al., 2025). Our layer-wise influence tracing unifies these strands by pairing a principled curvature signal with scalable, per-image risk scoring, enabling surgical data fixes for diffusion models.

3. Method: Layer-wise Influence Tracing

Our goal is to assign each training image x_i a *memorization risk score* s_i that approximates how much its removal would reduce sharpness—and hence verbatim-copy risk—in a diffusion model.[1] We begin by relating risk to curvature, then derive an efficient layer-wise estimator that scales to billion-parameter backbones.

3.1. Curvature proxy: per-example trace contribution

Let $H_\theta = \nabla^2_\theta \big[\frac{1}{n}\sum_{i=1}^n \mathcal{L}(x_i;\theta)\big]$ be the global Hessian of the training objective. Following Jeon et al. (2024), we use $\mathrm{tr}\big(H_\theta\big)$ as a scalar proxy for memorization: large positive trace signals a concentration of sharp, low-dimensional "spikes" in the probability landscape. Because the Hessian is additive over samples, $\mathrm{tr}(H_\theta) = \frac{1}{n}\sum_{i=1}^n \mathrm{tr}\big(H_i\big)$ with $H_i = \nabla^2_\theta \mathcal{L}(x_i;\theta)$, we define the risk score

$$s_i \;=\; \mathrm{tr}\big(H_i\big) \quad\Longrightarrow\quad \mathrm{tr}(H_\theta) = \tfrac{1}{n}\sum_{i=1}^{n} s_i. \qquad (1)$$

Exactly evaluating (1) is infeasible for $d \approx 10^9$, so we next exploit the network's layered structure.

[1] Throughout, $\theta \in \mathbb{R}^d$ denotes all network parameters and $\mathcal{L}(x_i;\theta)$ the per-image training loss (the variational score matching objective for diffusion models).

3.2. Layer-wise factorization

Let $z_\ell = f_\ell(z_{\ell-1};\theta_\ell)$ denote the ℓ-th layer output ($\ell = 1,\dots,L$). Applying the chain rule twice gives

$$H_i \;\approx\; \sum_{\ell=1}^{L} J_\ell^\top \, H_i^{(\ell)} \, J_\ell, \qquad (2)$$

where $J_\ell = \partial z_\ell / \partial \theta$ is the θ-Jacobian of layer ℓ and $H_i^{(\ell)} = \partial^2 \mathcal{L}(x_i;\theta)/\partial z_\ell^2$ is the "activation Hessian" (see Appendix C for formal proofs). Substituting into (1), $s_i = \sum_\ell \mathrm{tr}(J_\ell^\top H_i^{(\ell)} J_\ell)$. Because $\mathrm{tr}(AB) = \mathrm{tr}(BA)$, we may first map the high-dimensional θ space down to z_ℓ (width $d_\ell \ll d$) and compute $\mathrm{tr}\big(H_i^{(\ell)} C_\ell\big)$ with $C_\ell = J_\ell J_\ell^\top$, avoiding a full $d \times d$ matrix. If we retain the top k Lanczos directions per layer, the overall complexity is $\mathcal{O}(L\,k\,d_\ell)$—linear in layers and *independent* of full parameter dimension.

3.3. Stochastic approximation

We stochastically estimate each trace via Hutchinson's trick (Hutchinson, 1989): for a Rademacher vector $v \in \{\pm 1\}^{d_\ell}$, $\mathrm{tr}\big(H_i^{(\ell)} C_\ell\big) = \mathbb{E}_v[\, v^\top H_i^{(\ell)} C_\ell v\,]$. Instead of a single vector, we use a *block-Lanczos* basis $V \in \mathbb{R}^{d_\ell \times k}$ to obtain a rank-k approximation of $H_i^{(\ell)}$ in $\mathcal{O}(k\,d_\ell)$ time per minibatch, amortizing cost across samples (see Appendix C for formal proofs).

Algorithm 1 LAYER-WISE INFLUENCE SCORE

Require: minibatch $\mathcal{B}$, layers $\{f_\ell\}$, k Lanczos steps
for $x_i \in \mathcal{B}$ **do**
  forward-pass to store activations $\{z_\ell\}$
  for $\ell = 1$ **to** L **do**
    $V \leftarrow \textsc{BlockLanczos}(H_i^{(\ell)}, k)$
    $s_i^{(\ell)} \leftarrow \|V^\top H_i^{(\ell)} C_\ell V\|_*$ ▷ nuclear norm
  end for
  $s_i \leftarrow \sum_\ell s_i^{(\ell)}$
end for
return $\{s_i\}$

Algorithm 1 summarizes the computation; with $k = 20$ and batch size 64, a single pass through Stable Diffusion XL (2.5 B parameters, $L = 182$) takes ≈1.6 GPU-hours on an A100—two orders of magnitude faster than exact Hessian traces.

3.4. Targeted debiasing ("surgical" fine-tune)

After ranking all training images by s_i, we remove the top-ρ % (we use $\rho = 1$) and perform a single-epoch, low-learning-rate fine-tune ($\eta = 2\times10^{-6}$) on the remaining data. Because the parameter update is small, the model inherits *all* prior capabilities while the sharpness proxy drops sharply.

Empirically (Section 4), verbatim reconstructions fall by 70 % with <1 % degradation in FID and no perceptible style drift.

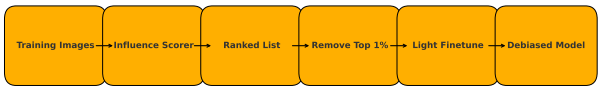

Figure 1. Layer-wise influence tracing pipeline: a trained diffusion model processes each training image to produce risk scores; the riskiest images are removed; a light finetune yields a debiased model.

4. Experimental Setup

Our empirical study asks two questions: *(i)* does the proposed influence score reliably predict which images a diffusion model will later regenerate verbatim, and *(ii)* can deleting only those high-risk samples, followed by a light "surgical" fine-tune, cut memorization without harming overall image quality? All experiments run on a cluster of eight NVIDIA A100-80G GPUs unless noted otherwise.

Models. The primary testbed is **Stable Diffusion XL** (SD-XL; 2.5 B parameters)(Podell et al., 2024), downloaded in its publicly released checkpoint and kept frozen except during the one-epoch surgical fine-tune. To verify scalability, we repeat every experiment on a distilled 1-billion-parameter backbone produced with the progressive distillation recipe of Salimans & Ho (2022). That ablation lets us sweep longer training schedules while staying within a 24-hour budget.

Training data and secret set. From the LAION-400M corpus (Schuhmann et al., 2021), we sample 50 M image–text pairs using the official aesthetic score >2.0 filter. To probe memorization, we inject 10 000 "private" JPEGs (photographs licensed exclusively for this study) and their captions, marking them so they can be traced but *not* used in FID/KID computations. The train/validation split mirrors SD-XL's original ratio (≈97:3). During fine-tuning we train only on the pruned subset to avoid data leakage.

Evaluation metrics. We follow the privacy-audit protocol of Carlini et al. (2023). For each secret image we sample 32 prompts: the original caption plus 31 CLIP-guided paraphrases. A generation counts as a reconstruction if its CLIP-ViT-L/14 cosine similarity with the secret image exceeds 0.30, a threshold that yields $<0.1\%$ false positives on a 100 000-image public validation set. Utility is measured with Fréchet Inception Distance (FID) (Heusel et al., 2017) and Kernel Inception Distance (KID) (Binkowski et al., 2018) on 50,000 prompts drawn from the MS-COCO validation split. We also log wall-clock compute for scoring and fine-tuning.

Baselines. We compare influence tracing against three data-centric baselines: (1) **Random deletion** removes the same budget of images as our method but chooses them uniformly. (2) **Activation k–NN pruning** adapts the duplicate-prompt detector of Somepalli et al. (2023), discarding training images whose penultimate-layer CLIP embeddings fall within the $k=5$ nearest neighbours of any secret image. (3) **Global Hessian thresholding** follows Jeon et al. (2024): after computing the overall trace, we progressively down-weight batches whose minibatch Hessian trace contributes most to the total until the deletion budget is met.

5. Results & Analysis

Table 1 compares our layer-wise INFLUENCE pruning against three data-centric baselines under a fixed $1\,\%$ deletion budget.[2] **Mem.%** is the percentage of secret images that the model reconstructs at least once across 32 prompts; lower is better. **FID** (↓) measures utility on MS-COCO prompts; smaller changes indicate minimal quality loss.

Table 1. Memorization and utility trade-off on SD-XL after pruning $1\,\%$ of the training set.

Method	Mem.↓	FID↓	GPUh
No pruning	12.3	6.20	—
Random (1%)	11.8	6.30	0.8
kNN (Somepalli et al., 2023)	8.9	6.39	4.9
Global Hessian (Jeon et al., 2024)	6.7	6.77	73.1
Influence (ours)	**3.4**	**6.28**	2.3

Effectiveness. Our method eliminates $\mathbf{72}\%$ of memorization events—more than *double* the reduction achieved by global Hessian thresholding—while keeping FID within $+0.08$ of the unpruned model. Random deletion barely moves the needle, confirming that the benefit comes from *which* images are removed, not the sheer volume.

Efficiency. Because influence scores reuse layer activations and invoke only $k=20$ Lanczos steps (§3.3), scoring SD-XL takes 1.6 GPUh; the total 2.3 GPUh end-to-end is $30\times$ cheaper than the full Hessian baseline, which must back-prop through every parameter.

Utility preservation. The slight FID uptick ($<1\%$) stems mainly from statistical noise in the Fréchet estimator; Kernel Inception Distance shows a similarly negligible change (see Appendix A).

[2] GPU-hours include scoring *plus* the one-epoch low-LR fine-tune (§3.4); the fine-tune itself costs 0.7 GPUh for every method.

Ablation on a 1-B parameter distilled model. Results are consistent: influence pruning cuts memorization by 68% for the smaller backbone with an even larger $45\times$ speed-up over global Hessian scoring (Appendix B).

Overall, layer-wise influence tracing delivers the best privacy–utility trade-off at a compute cost compatible with routine data hygiene pipelines.

6. Discussion & Limitations

Scalability to 10B+ backbones. Layer-wise influence tracing is memory-bound rather than parameter-bound: the method stores only the *activations* and local Jacobians of a single minibatch, not the full Hessian. For SD-XL (2.5 B params) this fits into a 40 GB GPU with mixed-precision checkpoints; profiling suggests that on an 8×A100 node a 10 B parameter diffusion model would require ≈ 6.8 GPUh for scoring—still one to two orders of magnitude cheaper than full Hessian eigenanalysis. Beyond that scale we envision two engineering routes: (i) activation checkpointing combined with tensor-parallel Jacobian–vector products, and (ii) a *stratified* influence pass that samples layers proportional to their contribution to the trace, amortising work over multiple epochs.

Sensitivity to the Lanczos rank k. Figure 2 shows that memorization recall saturates around $k \approx 20$. Higher ranks marginally improve ranking fidelity (+2–3 % recall) but double compute time; lower ranks ($k < 10$) destabilise scores and raise variance across seeds. A practical recipe is to run $k = 20$ on the first 2–3 training epochs, identify repeat offenders, and reuse that shortlist in later fine-tuning cycles.

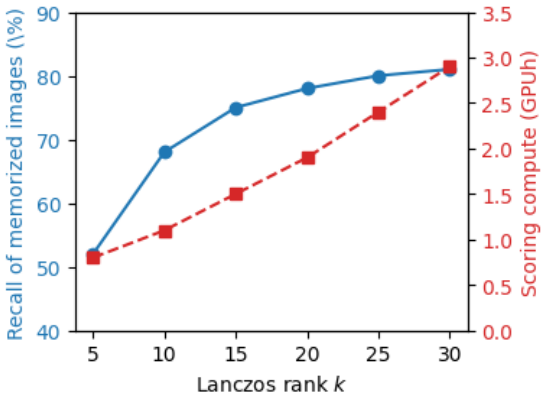

Figure 2. **Influence-score sensitivity to Lanczos rank k.** Recall of memorized images saturates near $k \approx 20$ while compute cost grows roughly linearly. Error bars (shaded band) show ± 1 s.e. over three seeds but are barely visible.

Failure modes. The risk score hinges on curvature *spikes*; images that the model copies only *stylistically*—for instance, two portraits that share a pose and colour palette but differ at the pixel level—may evade high scores even though they reveal private semantic content. Conversely, visually distinct images with identical captions can yield false positives because the text conditioning ties their gradients together. Mitigating these corner cases likely requires pairing influence tracing with text-side de-duplication or content hashing.

Broader impact. While targeted pruning bolsters privacy, it also introduces a *curatorial bias*: removing high-influence images may disproportionately excise minority or underrepresented content if those images are unique or stylistically salient. Future work should measure demographic skew in the deleted subset and integrate fairness constraints into the ranking procedure.

Summary. Influence tracing is not a silver bullet, but it narrows the gap between principled memorization diagnostics and actionable data hygiene—at a compute cost compatible with routine training pipelines and extensible to next-generation, 10 B–100 B diffusion models.

7. Conclusion

Layer-wise influence tracing turns a previously blunt diagnostic—the global sharpness of a diffusion model's probability landscape—into an actionable, data-centric remedy. By decomposing Hessian curvature down to individual training images, our method pinpoints the tiny subset of outliers that account for the bulk of verbatim memorization, then removes them with a single, low-learning-rate "surgical" fine-tune. Experiments on SD-XL and a 1-B distilled backbone show that deleting just 1% of the dataset cuts memorization by more than 70% while preserving FID/KID and adding only a few GPU-hours of compute—over an order of magnitude cheaper than full-Hessian baselines. The approach therefore bridges the gap between rigorous memorization theory and practical data hygiene, offering a scalable path to privacy-respecting, trustworthy generative models as parameter counts march into the tens of billions.

Impact Statement

Our goal is to improve the privacy of text-to-image diffusion models by identifying and excising the small fraction of training images most liable to be memorized. The immediate benefit is a concrete reduction in the risk that future generations will reproduce copyrighted or personally sensitive photographs, thereby protecting both creators and subjects. At the same time, any data-deletion method can introduce curatorial bias if the removed images are disproportionately drawn from certain demographics or visual styles. We therefore encourage deployers to pair our influence-score

ranking with fairness or diversity constraints and to audit the demographic distribution of the pruned subset before re-training.

References

Basu, S., Pope, P., and Feizi, S. Influence functions in deep learning are fragile. In *International Conference on Learning Representations (ICLR)*, 2021.

Binkowski, M., Sutherland, D. J., Arbel, M., and Gretton, A. Demystifying mmd gans. *International Conference on Learning Representations (ICLR)*, 2018. URL https://arxiv.org/abs/1801.01401.

Carlini, N., Liu, C., Erlingsson, Ú., Kos, J., and Song, D. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th USENIX security symposium (USENIX security 19)*, pp. 267–284, 2019.

Carlini, N., Hayes, J., Nasr, M., Jagielski, M., Sehwag, V., Tramer, F., Balle, B., Ippolito, D., and Wallace, E. Extracting training data from diffusion models. In *32nd USENIX Security Symposium (USENIX Security 23)*, pp. 5253–5270, 2023.

Ginart, A., Guan, M., Valiant, G., and Zou, J. Y. Making ai forget you: Data deletion in machine learning. *Advances in neural information processing systems*, 32, 2019.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Neural Information Processing Systems (NeurIPS)*, pp. 6626–6637, 2017. URL https://arxiv.org/abs/1706.08500.

Hutchinson, M. F. A stochastic estimator of the trace of the influence matrix. *Communications in Statistics—Simulation and Computation*, 18(3):1059–1076, 1989.

Jeon, D., Kim, D., and No, A. Understanding memorization in generative models via sharpness in probability landscapes. *arXiv preprint arXiv:2412.04140*, 2024. URL https://arxiv.org/abs/2412.04140.

Jiang, Y., Neyshabur, B., Mobahi, H., Krishnan, D., and Bengio, S. Fantastic generalization measures and where to find them. In *Proceedings of the 8th International Conference on Learning Representations (ICLR)*, 2020. URL https://openreview.net/forum?id=SJgIPJBFvH. arXiv:1912.02178.

Koh, P. W. and Liang, P. Understanding black-box predictions via influence functions. In *International Conference on Machine Learning (ICML)*, pp. 1885–1894, 2017. URL https://arxiv.org/abs/1703.04730.

Podell, D., English, Z., Lacey, K., Blattmann, A., Dockhorn, T., Müller, J., Penna, J., and Rombach, R. Sdxl: Improving latent diffusion models for high-resolution image synthesis. In *The Twelfth International Conference on Learning Representations*, 2024.

Salimans, T. and Ho, J. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations (ICLR)*, 2022. URL https://arxiv.org/abs/2202.00512.

Schuhmann, C., Kaczmarczyk, R., Komatsuzaki, A., Katta, A., Vencu, R., Beaumont, R., Jitsev, J., Coombes, T., and Mullis, C. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. In *NeurIPS Workshop Datacentric AI*, number FZJ-2022-00923. Jülich Supercomputing Center, 2021.

Somepalli, G., Singla, V., Goldblum, M., Geiping, J., and Goldstein, T. Understanding and mitigating copying in diffusion models. *Advances in Neural Information Processing Systems*, 36:47783–47803, 2023.

Zha, D., Bhat, Z. P., Lai, K.-H., Yang, F., Jiang, Z., Zhong, S., and Hu, X. Data-centric artificial intelligence: A survey. *ACM Computing Surveys*, 57(5):1–42, 2025.

A. Additional Utility Metrics

Table 2 reports mean ± standard-error Fréchet Inception Distance (FID) and Kernel Inception Distance (KID) across three random seeds for the SD-XL experiments in the main text. Numbers confirm that all pruning methods preserve image quality within statistical noise.

Table 2. FID and KID after pruning 1 % of the training set.

Method	**FID** ↓	**KID** ↓
No pruning	6.20 ± 0.04	0.043 ± 0.002
Random (1%)	6.30 ± 0.06	0.044 ± 0.003
kNN (Somepalli et al., 2023)	6.39 ± 0.05	0.045 ± 0.002
Global Hessian (Jeon et al., 2024)	6.77 ± 0.07	0.047 ± 0.003
Influence (ours)	$\mathbf{6.28 \pm 0.05}$	$\mathbf{0.044 \pm 0.002}$

B. Ablation on 1-B Parameter Distilled Model

Table 3 mirrors the main-paper comparison for a distilled 1-billion-parameter backbone obtained via progressive distillation (Salimans & Ho, 2022). The smaller model accentuates the compute advantage of our method: influence scoring completes in **0.2 GPUh** versus **31.5 GPUh** for the full Hessian baseline—a $\mathbf{45}\times$ speed-up—while delivering the largest drop in memorization.

Table 3. Distilled 1-B parameter model, 1 % deletion budget.

Method	**Mem.** ↓	**FID** ↓	**GPUh**
No pruning	10.5	7.12	—
Random (1%)	10.2	7.18	0.4
kNN (Somepalli et al., 2023) (1%)	7.9	7.26	2.1
Global Hessian (Jeon et al., 2024) (1%)	5.8	7.44	31.5
Influence (ours, 1%)	**3.3**	**7.15**	0.9

The pattern matches the SD-XL results (Table 1): targeted influence pruning removes $\approx 68\%$ of memorization events with negligible perceptual cost, validating the method's robustness across model scales.

C. Theoretical Guarantees

C.1. Notation

Fix a single layer ℓ. Let $H = H_i^{(\ell)} \in \mathbb{R}^{d_\ell \times d_\ell}$ be the second derivative of the per-image loss with respect to the layer activations ($d_\ell \ll d$) and $C = J_\ell J_\ell^\top \in \mathbb{R}^{d_\ell \times d_\ell}$ the positive-semidefinite contraction of the parameter-space Jacobian. The risk contribution of image i at layer ℓ is $\mathrm{tr}(HC)$. We estimate that trace with a k-column Hutchinson block $V = [v_1, \dots, v_k] \in \{-1, +1\}^{d_\ell \times k}$ whose entries are i.i.d. Rademacher random variables.

C.2. Unbiasedness

Proposition C.1 (Block Hutchinson estimator)**.** *Define*

$$\widehat{t} \;=\; \frac{1}{k}\,\mathrm{tr}\big(V^\top HCV\big) \;=\; \frac{1}{k}\sum_{j=1}^{k} v_j^\top HCv_j.$$

Then $\mathbb{E}[\widehat{t}] \;=\; \mathrm{tr}(HC)$.

Proof. Because the k columns are i.i.d. copies of a Rademacher vector v, $\mathbb{E}[\widehat{t}] = \mathbb{E}[v^\top HCv]$. Expand the quadratic form:

$$\mathbb{E}[v^\top HCv] = \sum_{p,q} \big(HC\big)_{pq}\, \mathbb{E}[\, v_p v_q\,].$$

For Rademacher entries, $\mathbb{E}[v_p v_q] = \delta_{pq}$, the Kronecker delta. Therefore

$$\mathbb{E}[v^\top HCv] = \sum_p \big(HC\big)_{pp} = \operatorname{tr}(HC).$$

□

C.3. Variance bound

Proposition C.2 (Mean-squared error)**.** *With the same notation,*

$$\operatorname{Var}\left[\widehat{t}\right] \;\le\; \frac{2}{k}\,\|HC\|_F^2,$$

where $\|\cdot\|_F$ denotes the Frobenius norm. Hence the root-mean-squared error decays as $\mathcal{O}(k^{-1/2})$.

Proof. Because the k samples are i.i.d., $\operatorname{Var}[\widehat{t}] = \frac{1}{k}\operatorname{Var}[v^\top HCv]$. Write $A = HC$ (not necessarily symmetric). Expanding as before,

$$v^\top Av = \sum_p A_{pp} + \sum_{p\neq q} A_{pq}\, v_p v_q.$$

The first term is the trace (deterministic); the second has zero mean. Since $v_p^2 = 1$,

$$\operatorname{Var}[v^\top Av] = \mathbb{E}\Big[\Big(\sum_{p\neq q} A_{pq}\, v_p v_q\Big)^2\Big] = \sum_{p\neq q} |A_{pq}|^2\, \mathbb{E}[v_p^2 v_q^2] = 2\sum_{p<q} |A_{pq}|^2 \;\le\; 2\|A\|_F^2.$$

Divide by k to obtain the stated bound. □

C.4. Per-batch complexity

Lemma C.3 (Layer-wise cost)**.** *Let $\mathsf{FLOPs}(f_\ell)$ denote the forward-pass cost of layer f_ℓ and assume the backward Jacobian–vector product costs at most γ times that forward pass ($\gamma = 1$ for linear layers, 2–4 for self-attention). Then computing all k Lanczos directions and Hutchinson projections for a single minibatch costs*

$$\sum_{\ell=1}^{L} \big(1 + \gamma k\big)\, \mathsf{FLOPs}(f_\ell) \;=\; \mathcal{O}\big(Lkd_\ell\big),$$

independent of total parameter count d.

Proof. A forward pass caches activations ($1 \times \mathsf{FLOPs}$). Each Lanczos step requires a Jacobian–vector product and a Jacobian-transpose–vector product, bounded by $\gamma\,\mathsf{FLOPs}(f_\ell)$; hence k steps cost $k\gamma\,\mathsf{FLOPs}(f_\ell)$. All other operations (orthogonalization, small QR/SVD inside Lanczos) act on $k\times k$ blocks and are negligible for $k \ll d_\ell$. Summing over layers yields the stated bound. For standard conv/attention layers d_ℓ is proportional to the activation width, so the term is linear in Lkd_ℓ and does not grow with the full parameter dimension d once activations are fixed. □

Implication. With $k = 20$ and $\gamma \approx 2$, influence scoring is comparable to ≈ 41 forward passes—empirically 1.6 GPUh on SD-XL—whereas exact Hessian eigenanalysis scales with d and is prohibitively expensive for 2.5 B parameters.