Mitigating Memorization Where It Happens

Xuanqi Zhang[1,2*] **Haoyang Shang**[1,2*] **Xiaoxiao Li**[1,2†]
[1]Department of Electrical and Computer Engineering,
The University of British Columbia
[2]Vector Institute

Abstract

Large language models (LLMs) can memorize and reproduce training sequences verbatim, raising privacy and copyright concerns. Existing mitigation methods apply interventions uniformly, often degrading predictions at the many positions unrelated to verbatim recall. We study a token-level log-likelihood ratio between a target model and a domain-competent reference with lower exposure to the evaluated sequences. Exposure-controlled experiments support its use for localizing positions associated with sequence-level verbatim recall. Across evaluated settings, a minority of positions carries a disproportionate share of the positive signal, motivating selective activation correction. From a first-order surrogate of the memorization-reduction objective, we derive a *probe–steer framework*, which decomposes intervention into a probe that flags high-signal activations and a steer that applies a correction only when the probe exceeds a threshold. We propose Gated Subspace Steering (GSS) as its practical instantiation: under a background normalization constraint, the probe–steer pairs that are optimal for this surrogate follow in closed form from the leading singular vectors of a whitened gradient-weighted activation matrix. Across four benchmarks, GSS achieves memorization suppression comparable to strong baselines, with favorable utility trade-offs in several settings, while requiring 10–100$\times$ less compute than optimization-based alternatives. Code is available at `https://github.com/ubc-tea/Mitigating-Memorization-Where-It-Happens`.

1 Introduction

Large language models (LLMs) can reproduce training sequences verbatim, creating risks of training-data extraction and disclosure of sensitive or copyrighted content [10, 55, 44, 68, 46]. Downstream adaptation is a standard use of pretrained language models [26]. Additional training on downstream data can increase such reproduction, particularly when sequences are repeated or heavily represented [64, 37]. Verbatim memorization can coexist with useful generalization: a model may perform well on unseen examples while retaining and reproducing particular training sequences. Our goal is to reduce this sequence-level reproduction while preserving the model's utility on other inputs. Motivated by the pitfalls of evaluating memorization [2], we treat memorization suppression and generalization performance as separate evaluation dimensions, and study how inference-time interventions can control their trade-off without updating model weights.

Existing approaches offer several ways to modify model behavior. *Machine unlearning* aims to remove the influence of designated training examples [5, 4, 23, 63], with methods such as NPO optimizing an alignment-inspired objective to mitigate the instability of gradient ascent [72]. *Knowledge editing* focuses on changing specific factual associations while preserving unrelated behavior [62, 19]. For example, SERAC uses an external memory and a routing mechanism to implement targeted edits [42].

*Equal contribution.
†Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Memorization mitigation directly evaluates whether an intervention reduces the reproduction of training sequences [11, 57, 12, 24]. Localization-based approaches identify and suppress neurons or parameters associated with such reproduction. These objectives overlap, and unlearning or editing methods can also be evaluated for memorization suppression. We include them as objective-related baselines and separately compare activation-space interventions to assess the role of selective steering.

Activation steering provides a lightweight mechanism for controlling model behavior by modifying hidden states during inference, without updating model parameters [39, 49, 25]. A common approach adds a fixed direction at every token position. For memorization mitigation, such uniform intervention may perturb useful predictions even where little correction is needed. We investigate selective intervention using the token-wise log-likelihood difference between the target model and a domain-competent reference with lower exposure to the evaluated sequences. Across our evaluated settings, a minority of positions accounts for a disproportionate share of the positive likelihood difference. Exposure-controlled analyses further associate this signal with sequence-specific recall, while showing that an individual high-signal position can also reflect legitimate learning. These findings motivate using the signal to guide selective activation correction, with its effectiveness assessed through sequence-level reproduction and independent utility metrics.

These observations motivate separating the decision of where to intervene from the choice of how to modify an activation. A probe can identify candidate intervention positions from the current hidden state, while a distinct steering direction specifies the correction. The target–reference likelihood signal guides offline calibration; during generation, intervention depends only on the learned activation probe and does not require evaluating the reference model. We consider both a single-position gate and a variant requiring threshold exceedance at two consecutive positions, and evaluate their effects on sequence-level reproduction and utility separately.

Building on this design, we propose **Gated Subspace Steering (GSS)**, an inference-time intervention combining a low-rank *probe*, a threshold *gate*, and a corrective *steer*. During calibration, the target–reference signal identifies high-signal positions whose activations and loss gradients guide the construction of probe–steer directions. A covariance-weighted linear surrogate yields a closed-form direction-finding procedure based on singular value decomposition. During generation, the gate selectively applies activation corrections using the calibrated directions. The theoretical analysis characterizes the local surrogate; sequence-level suppression and utility under the deployed intervention are evaluated empirically. Our contributions are as follows:

- We study the token-wise decomposition of target–reference excess log-likelihood, which serves as a signal for localizing positions associated with sequence-level verbatim recall. Exposure-controlled experiments validate its relationship to training exposure, and measurements across twelve model–dataset settings characterize the concentration of its positive mass.
- We introduce GSS, which separates activation-based probing from corrective steering and applies corrections through a configurable gate. We derive closed-form directions for a constrained first-order linear surrogate and distinguish this result from the behavior of the nonlinear intervention during generation.
- We evaluate memorization suppression and utility separately across pretrained and fine-tuned models. GSS achieves suppression comparable to strong baselines, with favorable utility trade-offs in several settings and low inference-time overhead. We further examine how gate configuration affects suppression and incidental changes to useful outputs.

2 Related work

2.1 Memorization in language models

Prior work shows that LLMs can memorize rare training examples and reproduce them verbatim at inference time [9, 15, 73, 32]. Carlini et al. [9, 8] propose the exposure metric to quantify how much more likely a model is to generate a specific sequence than expected under a random baseline. Subsequent work extends this analysis across architectures and scales [7, 50, 29, 74], identifying training-data duplication, model size, and training duration as key drivers of memorization. Memorization also varies across examples and evaluation settings [1, 14]. In parallel, black-box extraction attacks [10] demonstrate that memorization can lead to the recovery of verbatim training

data from models such as GPT-2 [48]. Beyond verbatim reproduction, Zeng et al. [71] study paraphrase and idea memorization. Our work primarily targets sequence-level verbatim reproduction

2.2 Activation steering and representation engineering

Activation steering modifies internal representations at inference time to control model behavior. A common approach is contrastive activation addition, which extracts a direction from differences between contrasting examples and applies an update of the form $h \leftarrow h + \alpha v$ [49, 39, 59, 56]. Related methods use representation directions to encode semantic attributes or task signals, while adaptive and conditional approaches vary interventions according to the input or activation state [36, 60, 25, 75, 38]. Benchmarking work further evaluates representation-based interventions across behaviors [66]. Two design choices are relevant to our setting: how the correction direction is obtained, and when the correction is applied. GSS derives separate probe and steer directions from an activation–gradient coupling under a covariance constraint, then uses a calibrated gate to control their application. We compare activation-space interventions because they share this mechanism of behavioral control, and evaluate their suitability for reducing sequence-level verbatim reproduction while preserving utility.

3 Localizing excess predictive advantage within sequences

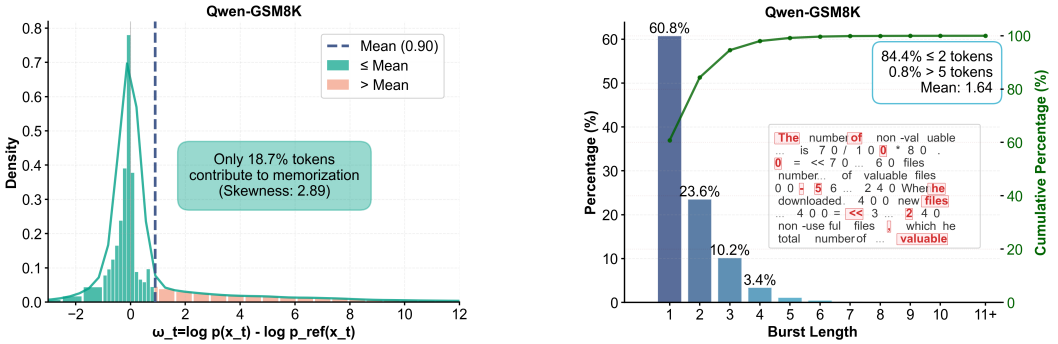
We evaluate memorization through sequence-level verbatim reproduction, using the standard notion of (n, k)-memorization:

Definition 3.1 ((n, k)-Memorization)**.** Let $s = (x_1, \ldots, x_n)$ be an n-token sequence that appears in the training set of a language model M. We say that s is (n, k)-*memorized* by M if, when prompted with the prefix $(x_1, \ldots, x_k)$, the model reproduces the remaining tokens $(x_{k+1}, \ldots, x_n)$ exactly under greedy decoding.

This definition specifies the behavior we seek to mitigate. To guide selective intervention, we additionally examine how the target model's predictive advantage over a reference is distributed within a sequence. The reference should capture broadly reusable linguistic and domain regularities while having lower exposure to the evaluated sequences. It need not be more accurate than the target or be its immediately preceding checkpoint.

For a sequence $x_{1:T}$, define the excess log-likelihood

$$S(x_{1:T}) = \log p_\theta(x_{1:T}) - \log p_{\text{ref}}(x_{1:T}) = \sum_{t=1}^{T} \omega_t. \tag{1}$$



(a) Distribution of the token-level signal.

(b) Lengths of positive-signal runs.

Figure 1: **Token-level excess likelihood on Qwen3-GSM8K.** (a) The distribution of ω_t is right-skewed (skewness 2.89), with some positions exhibiting much larger target–reference likelihood differences than others. (b) Consecutive runs of positions with $\omega_t > 0$ have a mean length of 1.64; 60.8% of runs contain one position. These statistics describe the likelihood signal, rather than the lengths of memorized spans or ground-truth transitions between recall and generalization. Cross-setting concentration measurements are reported in Appendix C.1.

This signed quantity measures the target's model-based code-length advantage over the reference, connecting to information-theoretic treatments of memorization [43]. Its token-wise decomposition identifies where that advantage arises; it does not assign ground-truth memorization labels to individual tokens.

Definition 3.2 (Token-Level Excess Log-Likelihood Signal)**.** Given a sequence $x_{1:T}$, the *token-level excess log-likelihood signal* at position t is

$$\omega_t \triangleq \log p_\theta(x_t \mid x_{<t}) - \log p_{\mathrm{ref}}(x_t \mid x_{<t}), \tag{2}$$

where p_θ and p_{ref} are the target and reference models, respectively.

The reference helps distinguish exposure-related gains from predictability shared by both models. For descriptive analysis, we define the positive signal mass as

$$m_t = \max(0, \omega_t), \qquad \mathcal{M}_+(x) = \sum_{t=1}^{T} m_t. \tag{3}$$

Unlike the signed sequence-level gap $S(x)$, $\mathcal{M}_+(x)$ retains positive contributions without cancellation by negative ones. For example, suppose the training set contains "*Alice's phone number is 555-0142*" and the target reproduces it. If both models predict "is" after "phone number" with similar high probability, then $\omega_t \approx 0$: this position contributes little to $\mathcal{M}_+(x)$ despite the target's high confidence. If the target has learned the particular number through exposure while the reference has not, the number's tokens can receive substantially higher probability under the target and contribute strongly to $\mathcal{M}_+(x)$. Target likelihood alone does not distinguish these two sources of predictability; the reference provides a comparative baseline. We use *high-signal* and *low-signal positions* to describe the resulting partition.

Concentration of positive signal mass. Figure 1 illustrates the distribution of the token-level signal and the lengths of its positive-signal runs on Qwen3-GSM8K. Across twelve model–dataset settings comprising 11,833 sequences and 1.72 million scored positions, the top 10% of positions account for 30%–89% of the total positive signal mass. The Gini coefficient ranges from 0.50 to 0.91. This concentration does not imply that positive-signal positions are rare: 47%–94% of positions have $\omega_t > 0$, but many contribute only small positive values. We therefore characterize the observation as concentration of positive excess likelihood, without assuming a particular heavy-tailed distribution. These measurements motivate selective intervention.

Relationship to training exposure. We validate the signal using controlled comparisons in which sequence inclusion determines exposure. On seed-controlled canary pairs, ω_t predicts positions with top-decile inclusion gain with an AUROC of 0.99. In a cross-model evaluation, scores from a Pythia-2.8B pair predict independently measured Pythia-6.9B inclusion-gain labels with an AUROC of 0.95, compared with 0.48 for target confidence. On natural Pile text, positive-signal positions are 7.7 times as likely as the remaining positions to occur within a 20-token verbatim reproduction. These results support exposure-sensitive localization (Appendix C.2).

Implications for intervention. An isolated high-signal position can reflect legitimate learning, so its suppression need not improve sequence-level memorization. Moreover, the offline likelihood signal and the activation probe used during generation are distinct quantities. We therefore evaluate the deployed gate directly, including whether requiring threshold exceedance at consecutive positions reduces incidental intervention. This motivates GSS as a selective activation intervention whose benefits and costs are measured through sequence-level reproduction and independent utility metrics.

4 Gated subspace steering

Section 3 shows that positive excess likelihood is concentrated in a minority of positions. Gated Subspace Steering (GSS) exploits this by intervening only where needed. It acts on the hidden state $h_t \in \mathbb{R}^d$ at a single chosen layer and has two stages (Fig. 2). An offline stage (Fig. 2a) estimates K pairs of directions: a *probe* u_k that decides *where* to intervene and a *steer* v_k that decides *how*. An online stage (Fig. 2b) applies a correction along v_k only when the response to u_k exceeds a threshold. We formulate the direction-finding problem in Section 4.1, solve it in closed form in Section 4.2, and then describe the gated intervention used during generation.

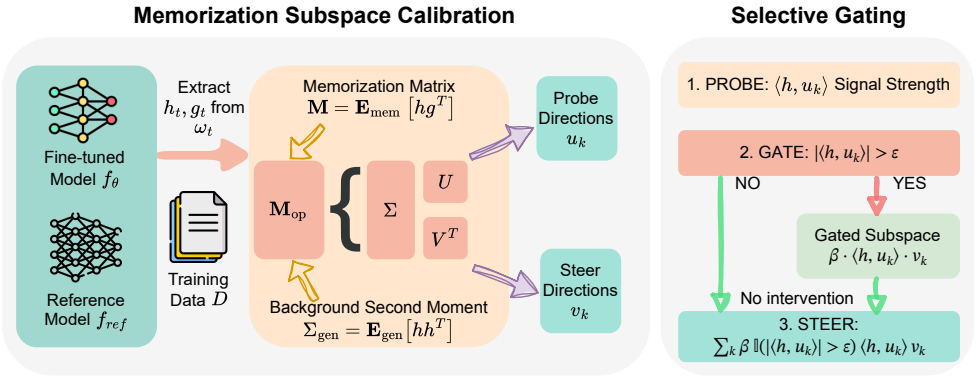


Figure 2: **Gated Subspace Steering overview. (a) Direction estimation.** The target–reference signal partitions positions into high- and low-signal subsets. Activation–gradient statistics from high-signal positions and a second moment from low-signal positions determine probe (u_k) and steer (v_k) directions through whitening and SVD. **(b) Intervention.** During generation, the gate applies a correction when the magnitude of a probe response exceeds its threshold. The response controls the signed correction along the steer. Gate decisions are not ground-truth memorization labels.

4.1 Problem formulation

We run the target and reference models on a set of training sequences $\mathcal{D}$ and compute ω_t (Eq. 2) at every position. This splits positions into a high-signal set $\mathcal{H} = \{t : \omega_t > 0\}$ and a low-signal background set $\mathcal{L} = \{t : \omega_t \leq 0\}$ (left of Fig. 2a). During generation, neither the reference model nor gradients are available, so the correction must be computed from h_t alone. We use a correction that reads the hidden state along a probe u and writes along a steer v:

$$\Delta h_t = -\beta\,(u^\top h_t)\,v, \qquad \beta > 0. \tag{4}$$

We keep u and v separate because recognizing a high-signal state and changing the next-token prediction are different tasks: the first depends on how activations are distributed, the second on how the output responds to them, and the two directions need not coincide.

Objective. The correction should lower the target's likelihood on high-signal positions. Let $\ell_{\mathcal{H}} = \frac{1}{|\mathcal{H}|}\sum_{t\in\mathcal{H}} \log p_\theta(x_t \mid x_{<t})$ be the mean log-likelihood on $\mathcal{H}$, and let $g_t = \nabla_{h_t}\ell_{\mathcal{H}}$ be its gradient with respect to the hidden state.[3] To first order, applying Eq. (4) at every $t \in \mathcal{H}$ changes $\ell_{\mathcal{H}}$ by

$$\Delta\ell_{\mathcal{H}} \approx \sum_{t\in\mathcal{H}} g_t^\top \Delta h_t = -\beta \sum_{t\in\mathcal{H}} (u^\top h_t)(g_t^\top v) = -\beta\, u^\top M_{\text{hi}}\, v, \qquad M_{\text{hi}} = \sum_{t\in\mathcal{H}} h_t g_t^\top. \tag{5}$$

Maximizing $u^\top M_{\text{hi}} v$ therefore maximizes the first-order likelihood reduction on high-signal positions. The *memorization matrix* M_{hi} (Fig. 2a, top) is large along pairs (u, v) for which the probe responds strongly exactly where moving along the steer lowers the likelihood.

Background constraint. Maximizing $u^\top M_{\text{hi}} v$ alone favors probes with large responses everywhere, including on background positions, where the gate would then fire. We prevent this by fixing the scale of the probe response on $\mathcal{L}$ using the background second moment (Fig. 2a, bottom):

$$\Sigma_{\text{lo}} = \frac{1}{|\mathcal{L}|}\sum_{t\in\mathcal{L}} h_t h_t^\top, \qquad C = \Sigma_{\text{lo}} + \gamma I, \tag{6}$$

with a small $\gamma > 0$ for numerical stability. Since $u^\top \Sigma_{\text{lo}} u$ is the mean squared probe response on background positions, requiring $u^\top C u = 1$ makes response scales comparable across probes, so a single threshold ϵ can be read in units of the typical background response.

Stacking $U = [u_1, \ldots, u_K]$ and $V = [v_1, \ldots, v_K]$, we seek K probe–steer pairs that solve

$$\max_{U,V} \operatorname{tr}\big(U^\top M_{\text{hi}} V\big) \quad \text{s.t.} \quad U^\top C U = I_K,\ \ V^\top V = I_K, \tag{7}$$

[3] g_t is the negative of the cross-entropy gradient. Batching and gradient normalization are described in Appendix A.3.

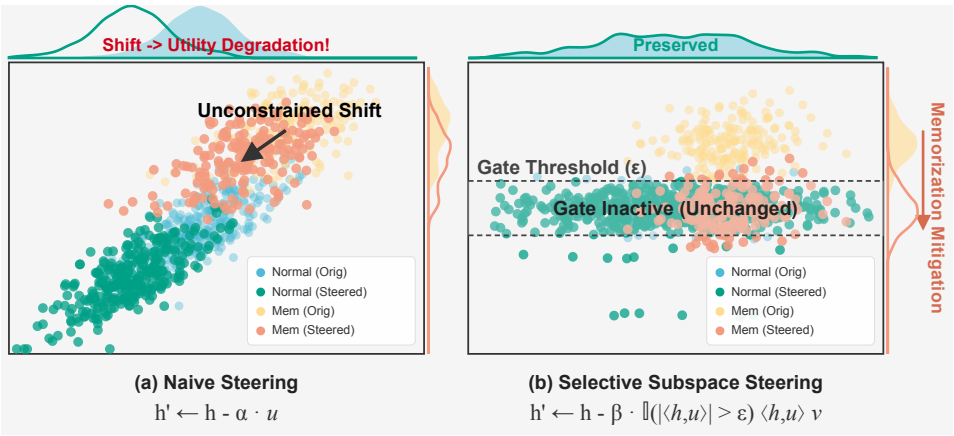


Figure 3: **Fixed and gated activation interventions.** (a) A fixed steering update is applied at every position. (b) GSS applies its correction only where a probe response exceeds the threshold. The illustrated regions are schematic and do not establish a separation between memorized and generalized computation.

where the first constraint bounds probe responses on the background and the second keeps the K steers orthonormal, preventing them from collapsing onto the same component.

4.2 Optimization

Although Eq. (7) is non-convex, it has an exact closed-form solution, connecting to spectral representation methods [3, 47].

Theorem 4.1 (Optimal whitened directions)**.** *Let $C = LL^\top$ be the Cholesky decomposition of C and let $L^{-1}M_{\text{hi}} = PSQ^\top$ be an SVD with singular values in descending order. Then $U^* = L^{-\top}P_{:,1:K}$ and $V^* = Q_{:,1:K}$ solve Eq. (7), with optimal value $\sum_{k=1}^{K}\sigma_k(L^{-1}M_{\text{hi}})$.*

Proof sketch. Substituting $\tilde{U} = L^\top U$ turns the first constraint into $\tilde{U}^\top\tilde{U} = I_K$ and the objective into $\text{tr}(\tilde{U}^\top L^{-1}M_{\text{hi}}V)$. By the von Neumann trace inequality, this is maximized by the leading K left and right singular vectors of $L^{-1}M_{\text{hi}}$; mapping back gives U^*. The full proof is in Appendix A.4. □

The solution corresponds to the middle of Fig. 2a and takes three steps: (i) whiten the memorization matrix by the background second moment, $L^{-1}M_{\text{hi}}$; (ii) compute its SVD; (iii) take the top-K left singular vectors, mapped back by $L^{-\top}$, as probes and the top-K right singular vectors as steers. The procedure requires no iterative optimization and no update to the model weights; its one-time cost is dominated by the SVD of a $d \times d$ matrix.

4.3 Gated intervention

During generation, GSS keeps (U^*, V^*) fixed and performs the three steps of Fig. 2b at each position:

$$\underbrace{s_{t,k} = u_k^\top h_t}_{\text{1. probe}}, \qquad h'_t = h_t - \beta \sum_{k=1}^{K} \underbrace{\mathbb{I}(|s_{t,k}| > \epsilon)}_{\text{2. gate}} \underbrace{s_{t,k}\, v_k}_{\text{3. steer}}. \tag{8}$$

The *probe* measures how strongly h_t expresses mode k. The *gate* activates mode k only when the magnitude of this response exceeds ϵ. The *steer* subtracts a correction along v_k scaled by the signed response. A run requirement r optionally restricts the gate to persistent responses: with $c_t = c_{t-1} + 1$ if the gate of the primary mode is active at position t and $c_t = 0$ otherwise, every correction term at t is multiplied by $\mathbb{I}(c_t \geq r)$. The single-position gate is $r = 1$; $r = 2$ intervenes from the second consecutive exceedance onward.

Fig. 3 contrasts this with fixed steering, which applies the same update at every position (Fig. 3a). A fixed update moves all activations, including background ones that need no correction, and shifts the model's overall behavior. GSS (Fig. 3b) applies the correction only above the threshold and leaves

the remaining positions untouched. We set ϵ to the $(1-\eta)$ quantile of probe responses on text, so η is the nominal fraction of positions at which a probe fires (Appendix D.5). At inference, GSS needs neither the reference model nor gradients, adding only $O(Kd)$ inner products per token at the intervened layer. Theorem 4.1 characterizes the first-order, ungated surrogate of Eq. (7). A large probe response marks a candidate position.

Contrastive configurations. Eq. (8) and Theorem 4.1 cover the whitened configuration (W). The contrastive configurations take the leading singular pair of the difference between the high- and low-signal activation–gradient matrices without whitening, accumulated at every position (C) or only at the masked positions (C-M; Appendix A.3), and apply the one-sided update $h'_t = h_t - \beta\,[u^\top h_t - \epsilon]_+\, v/\max(u^\top v, 10^{-3})$ (Appendix A.7).

5 Experiments

5.1 Experimental setup

We evaluate GSS across four settings spanning memorization-mitigation benchmarks, scalability tests, and realistic fine-tuning scenarios: (i) **TinyMem** [57], a benchmark suite designed specifically for memorization mitigation; (ii) **Pythia-2.8B and Pythia-6.9B** on their pretraining corpus, to assess scalability; (iii) **Llama-3.2-1B and Qwen3-0.6B fine-tuned on GSM8K** (reasoning) [67] and **UltraChat** (dialogue) [17]. We apply GSS at layer 4 for TinyMem models and layer 18 for Pythia models (Layer Selection Ablation is provided in Appendix F.3). **Gate configurations.** Except in Section 5.6, all results in this section use the single-position gate ($r = 1$). Table 1 uses whitened directions and Table 2 contrastive directions (Appendix A.7), both with rank one. Table 4 lists the configuration of every result that reports $r = 2$.

Baselines. We organize baselines by intervention target. *Parameter-space* methods modify model weights θ and include three families: *unlearning* (GA [33], NPO [72]) and localization-based (Greedy, SOU, Dura, Subnet, BalSub [51]), plus the LoRA-based O^3 [21]; *model editing* (ROME [40], MEMIT [41]); and *task arithmetic* (TaskVector [31]). *Inference-time* methods operate on activations h without weight updates: feature-suppression methods (Zero, Slimming, Hard Concrete, IG [11]) and steering methods (LUNAR [53], AlphaSteer [54], LEACE [3]), to which our method belongs.

Metrics. We report **Memorization** via (n, k)-memorization [57, 10], adversarial extraction and approximate memorization results are in Appendix E.5, **Utility** via Perplexity on held-out test sets and downstream task Accuracy and **Efficiency** via wall-clock time of the inference stage.

5.2 Experimental results on TinyMem

As shown in Table 1, our method yields zero observed reproduction under the evaluated protocol in all TinyMem settings. While some baselines reach zero memorization in isolated cases, they are often at the cost of catastrophic utility loss. Neuron- and weight-level pruning methods (HC, Slim, Act) also suppress memorization but substantially degrade accuracy, particularly on math tasks that require precise algorithmic structure. In contrast, our method preserves near-baseline accuracy on math models and stable perplexity on language models, demonstrating selective suppression without disrupting learned computation. Backdoor settings further highlight the limitations of static interventions. Many baselines that perform well under noise fail completely under backdoor perturbations, often reverting to 100% memorization (e.g., HC, Slim, Act, Zero, Greedy on language backdoor). In contrast, our method remains effective under backdoor conditions. Moreover, our method introduces a negligible inference-time overhead, making it suitable as a lightweight deployment-time safeguard.

5.3 Experimental results on Pythia

On pretrained Pythia models, GSS reduces (n, k)-memorization from 89.31% to 6.96% on Pythia-6.9B and from 52.87% to 6.93% on Pythia-2.8B, with test perplexity rising from 19.46 to 29.15 ($+49.8\%$) and from 21.75 to 28.26 ($+29.9\%$), respectively. This perplexity cost is larger than in fine-tuning settings because pretrained models lack a clean forget/retain split, leaving less margin for utility preservation. Among the evaluated methods, those matching GSS on perplexity (HC, Subnet, BalSub) leave memorization above 84%, while those approaching GSS on memorization (Dura, Dura-agg) inflate perplexity by 20%–77% and require more wall-clock time.

Table 1: **TinyMem Models** [51]. Comparison of memorization mitigation strategies on multiplicative math and language models under noise and backdoor settings. The results are averaged with 3 seeds. Metrics include (n, k)-memorization, test accuracy / perplexity, and wall-clock time of the mitigation pipeline (seconds). GSS uses whitened directions and the single-position gate ($r = 1$).

Method	Year	Math Model						Language Model					
		Noise			Backdoor			Noise			Backdoor		
		%Mem ↓	Acc ↑	Time ↓	%Mem ↓	Acc ↑	Time ↓	%Mem ↓	Perp ↓	Time ↓	%Mem ↓	Perp ↓	Time ↓
Baseline	–	34.55	97.00	–	99.44	96.81	–	17.60	57.73	–	100.00	58.45	–
Spectral reg	2024	0.17	96.84	12254	99.89	96.68	1131.5	0.00	57.92	10850.2	100.00	90.42	238.6
Loss trunc	2024	30.58	96.97	2391	99.67	99.96	347.7	0.00	57.92	16080	100.00	58.45	6005
Both FT	2024	0.00	97.17	28.45	0.00	91.36	20.35	0.00	51.63	1680	0.00	59.19	1629
Clean FT	2024	0.00	59.66	2.84	0.00	70.71	2.91	0.00	70.41	6.35	0.00	77.19	6.36
Extra FT	2024	0.00	73.22	27.33	0.00	74.63	18.58	0.00	52.00	1670	0.00	63.60	1624
HC	2024	0.00	74.97	0.24	0.00	75.02	0.26	0.00	57.15	0.25	100.0	70.94	0.69
Slim	2024	0.00	63.34	1.47	0.00	57.12	1.41	0.00	57.10	0.42	100.0	70.40	0.89
Act	2024	0.00	60.22	0.37	0.00	78.89	0.27	0.00	71.32	0.37	100.00	70.98	0.43
IG	2024	24.30	96.98	2657	46.67	54.19	1586	0.00	57.12	1647	0.00	102.7	1612
Zero	2024	22.15	96.86	11.91	99.44	73.44	12.10	0.00	57.12	16.47	100.0	70.44	63.05
Greedy	2024	12.73	96.19	17.62	46.67	54.19	18.46	0.00	102.7	1612	100.0	119.3	217.9
SOU	2024	0.81	87.02	313.4	78.89	76.11	449.8	26.00	57.28	573.7	100.0	70.74	585.1
Dura	2024	13.72	96.45	2.78	0.00	65.63	3.23	2.80	57.52	0.73	100.0	73.43	3.36
Dura-agg	2024	14.71	96.41	2.60	24.69	86.18	3.08	0.40	68.47	0.75	100.0	85.14	3.33
Subnet	2024	20.33	96.91	0.48	0.00	28.66	0.46	14.80	57.28	0.41	94.87	70.83	1.00
BalSub	2025	0.66	91.97	7.29	0.00	92.01	6.96	0.80	57.14	299.5	0.00	71.86	917.9
TaskVector	2022	0.00	89.31	0.37	9.44	93.20	0.37	16.80	57.71	0.45	100.00	70.41	0.67
LEACE	2023	0.00	74.09	0.48	0.00	77.75	1.30	13.20	65.63	0.69	100.00	65.12	1.70
NPO	2024	0.56	27.26	0.57	0.00	4.88	0.56	0.00	229.45	1.28	0.00	858.15	1.38
0^3	2025	5.12	95.37	1.83	66.67	61.23	2.32	56.52	13.60	0.49	73.09	70.00	1.14
AlphaSteer	2025	24.30	96.98	6.35	0.92	61.24	2.09	0.00	64.29	15.82	0.00	76.08	18.74
Lunar	2025	24.28	96.97	1.15	0.00	58.69	1.27	0.00	67.36	14.93	0.00	75.96	17.61
Ours	2026	0.00	96.98	0.001	0.00	96.82	0.001	0.00	63.13	0.002	0.00	63.17	0.003

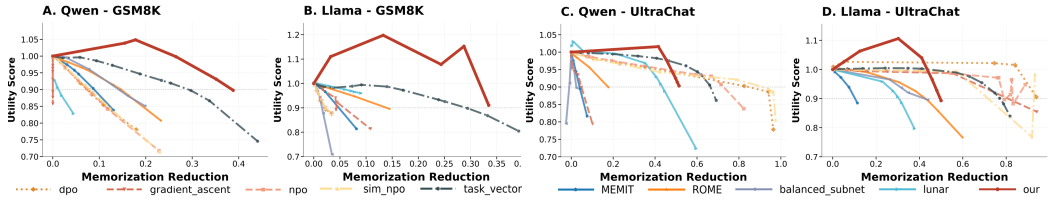


Figure 4: **Pareto Frontier Analysis.** The plots visualize the trade-off between memorization reduction measured and downstream utility measured in log-likelihood. The ideal method occupies the top-right corner. On reasoning tasks, GSS keeps the highest utility over the reduction range it reaches. On instruction tasks, GSS keeps higher utility than the *Editing* baselines over its range.

5.4 Pareto frontier analysis on modern LLMs

To evaluate the efficiency of our intervention, we conduct a comprehensive Pareto analysis. Specifically, we sweep the *learning rate* for Unlearning methods (e.g., GA), the *regularization weight* for localization methods (e.g., BalSub), and the *intervention magnitude* for Editing methods (e.g., ROME). Fig. 4 shows the resulting trade-off curves. GSS covers a smaller range of memorization reduction than the strongest baselines and keeps high utility within that range.

Reasoning Tasks. On complex reasoning benchmarks (first two in Fig. 4), we observe a clear hierarchy. *Unlearning methods* (e.g., GA) struggle significantly. *Knowledge Editing methods* (ROME, MEMIT) show better stability but hit a performance ceiling. GSS keeps the highest utility up to its largest reduction, with a slight **utility boost** (scores exceeding 1.0) at moderate steering strengths, while task vectors reach larger reductions at lower utility. This suggests that for reasoning tasks, memorization acts as overfitting noise; by selectively projecting it out via a gated subspace, we recover latent generalization capabilities on the test set.

Table 3: Single-position ($r = 1$) and two-consecutive-position ($r = 2$) gates with the same directions, ϵ, and β. Reproduction is the first 20 generated tokens (Pythia), the first 30 answer tokens (GSM8K), the first 50 continuation tokens (domains), or the full record (PII). All rows use $\eta = 0.1$.

			Reproduction ↓			Utility		
Model	Setting	β	None	$r = 1$	$r = 2$	None	$r = 1$	$r = 2$
Pythia-2.8B	Pile	10	0.665	0.142	0.175	22.35	24.90	24.03
Qwen2.5-7B	GSM8K	3	0.997	0.538	0.848	0.982	0.777	0.940
Qwen2.5-7B	Healthcare	10	0.913	0.027	0.600	1.089	20.885	1.768
Qwen2.5-7B	Finance	10	0.933	0.054	0.564	1.098	22.502	1.776
Qwen2.5-7B	Academia	10	0.919	0.094	0.477	1.464	3.006	1.814
Qwen2.5-7B	PII, 3 copies	10	0.77	0.01	0.35	0.610	0.220	0.467
Qwen2.5-7B	PII, 8 copies	10	1.00	0.84	1.00	0.600	0.473	0.570

Dialogue Tasks. For dialogue generation (the last two panels in Fig. 4), Within the range of memorization reduction attained by GSS, it maintains higher utility than the *Knowledge Editing* baselines, with utility scores remaining above 1.0 at moderate reductions. Unlearning methods extend to larger reductions than GSS, although they have lower utility. This comparison also highlights the distinction between preventing exact reproduction and preserving the quality of open-ended responses.

Table 2: Comparison on **Pythia** models across (n, k)-memorization, test perplexity, and time.

	Pythia 2.8B			Pythia 6.9B		
Method	%Mem ↓	PPL ↓	Time ↓	%Mem ↓	PPL ↓	Time ↓
Baseline	52.87	21.75	–	89.31	19.46	–
HC	44.55	21.35	2.25	87.55	17.56	36.01
Slim	14.46	23.41	1.96	33.46	18.90	280.35
Act	14.46	24.43	13.01	30.30	18.34	36.02
Dura	6.93	35.49	17.44	14.65	23.34	44.34
Dura-agg	7.52	32.35	231.43	10.98	34.40	320.60
Subnet	8.01	32.20	48.75	84.95	17.16	519.44
BalSub	5.94	31.52	2068.77	86.73	17.15	233.42
Ours	**6.93**	**28.26**	**0.16**	**6.96**	29.15	**0.21**

5.5 Gated vs. ungated intervention

We isolate the contribution of the gating mechanism by comparing GSS to its ungated variant: same calibrated (u^*, v^*), but the indicator $\mathbb{I}(\cdot)$ is removed and the steer is applied uniformly to all tokens. Fig. 5 shows the comparison on TinyMem under varying noise levels: gated intervention achieves 2.4–4.1 PPL points lower than ungated, while both maintain 0.0% memorization. Table 18 in Appendix D.6 lists the exact values. Together, these results isolate the gating mechanism as the source of GSS's utility-preservation. By restricting corrections to positions with high probe responses, the gate limits unnecessary intervention at other positions while retaining substantial suppression of verbatim reproduction.

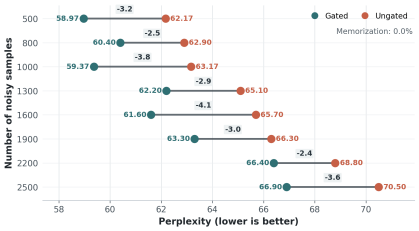


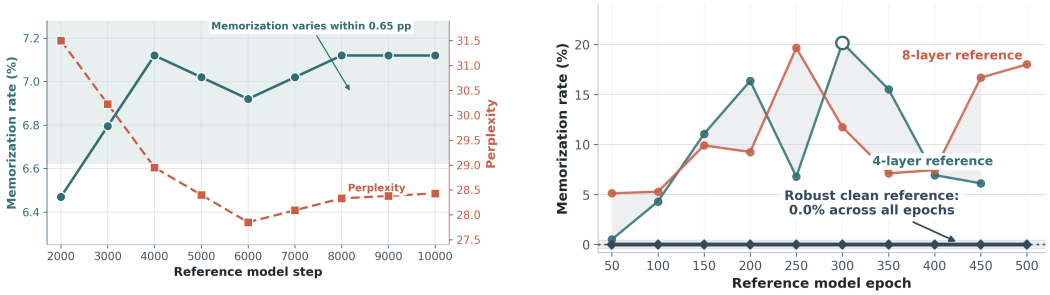
Figure 5: **Gated vs. Ungated.** Gating exploits heavy-tailed sparsity.

5.6 Single-position and consecutive gates

Table 3 compares $r = 1$ and $r = 2$ with identical directions, threshold, and strength in each setting. The consecutive gate fires less often and costs less utility, and the share of suppression it keeps depends on the setting: most of it on Pythia-2.8B, part of it on the fine-tuned Qwen2.5-7B tasks, and none with eight PII copies. At matched trigger rates on Pythia-2.8B, $r = 2$ reduces verbatim reproduction by 49.0 points against 45.4 for a single-position gate with a raised threshold, at equal correct-token alteration (3.34% and 3.33%). We therefore recommend $r = 2$ where incidental suppression is the main concern, and Table 3 shows the share of suppression it gives up in each setting (Appendix D).

5.7 Ablation study on rank choices

We analyze the impact of subspace rank k on the mitigation and utility trade-off in Fig. 7. The rank determines how many memorization-related directions are included in the steering subspace. All



(a) Sensitivity of Reference Model on Pythia-2.8B. (b) Sensitivity of Reference Model on TinyMem.

Figure 6: **Reference model ablation.** GSS is robust to the choice of reference model. Across reference checkpoints, the memorization rate remains around 6.5–7.1%.

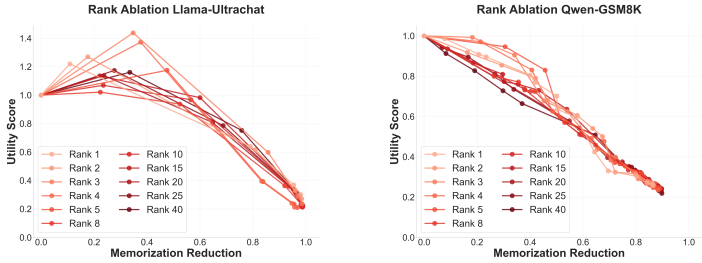


Figure 7: **Visualize the Pareto frontier**. Utility score vs. memorization reduction ($r = 1$).

ranks perform similarly in the low-utility region, and **lower ranks** ($k < 5$) keep higher utility in the high-utility regime. This pattern is consistent with memorization-related directions occupying a low-dimensional subspace, and it favors compact steering over high-rank intervention.

5.8 Reference model ablation

As shown in Figure 6, changing the reference checkpoint from 2k to 10k steps only slightly changes the final memorization rate, which remains around 6.5–7.1%. This suggests that GSS does not require a tuned reference checkpoint. We also compare noisy and clean references. Clean references consistently give near-zero memorization. Overall, GSS is robust to the choice of reference model.

6 Conclusion

We study selective activation interventions for reducing LLM memorization. A target–reference likelihood signal guides offline calibration, while learned activation probes control corrections during generation. GSS combines closed-form spectral calibration with configurable gates and leaves model weights unchanged. Our experiments demonstrate substantial memorization suppression with setting-dependent utility costs. Signal validity depends on reference suitability, and gate configuration changes the suppression–utility trade-off.

Acknowledgments and Disclosure of Funding

This work is supported, in part, by the Natural Sciences and Engineering Research Council of Canada (NSERC) through the Discovery Grant (RGPIN-2022-05316) and Alliance Grant (ALLRP 602633-24); Mitacs (IT34007); the IITP grants (RS-2024-00445087, RS-2025-25464461); the Gemini Research Program; the Amazon Research Award; the NVIDIA Academic Grant Program; the Terry Fox Research Institute and Health Research BC; the Canada CIFAR AI Chairs program; and the Canada Research Chair program. Furthermore, we thank UBC Advanced Research Computing and the Digital Research Alliance of Canada for providing invaluable computational resources. The authors declare no competing interests.

References

[1] George-Octavian Barbulescu and Peter Triantafillou. To each (textual sequence) its own: Improving memorized-data unlearning in large language models. *arXiv preprint arXiv:2405.03097*, 2024.

[2] Reza Bayat, Mohammad Pezeshki, Elvis Dohmatob, David Lopez-Paz, and Pascal Vincent. The pitfalls of memorization: When memorization hurts generalization. *arXiv preprint arXiv:2412.07684*, 2024.

[3] Nora Belrose, David Schneider-Joseph, Shauli Ravfogel, Ryan Cotterell, Edward Raff, and Stella Biderman. Leace: Perfect linear concept erasure in closed form. *Advances in Neural Information Processing Systems*, 36:66044–66063, 2023.

[4] Lucas Bourtoule, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *2021 IEEE symposium on security and privacy (SP)*, pages 141–159. IEEE, 2021.

[5] Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In *2015 IEEE symposium on security and privacy*, pages 463–480. IEEE, 2015.

[6] Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. Membership inference attacks from first principles. In *2022 IEEE symposium on security and privacy (SP)*, pages 1897–1914. IEEE, 2022.

[7] Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. Quantifying memorization across neural language models. In *The Eleventh International Conference on Learning Representations*, 2022.

[8] Nicholas Carlini, Matthew Jagielski, Chiyuan Zhang, Nicolas Papernot, Andreas Terzis, and Florian Tramer. The privacy onion effect: Memorization is relative. *Advances in Neural Information Processing Systems*, 35:13263–13276, 2022.

[9] Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th USENIX security symposium (USENIX security 19)*, pages 267–284, 2019.

[10] Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data from large language models. In *30th USENIX security symposium (USENIX Security 21)*, pages 2633–2650, 2021.

[11] Ting-Yun Chang, Jesse Thomason, and Robin Jia. Do localization methods actually localize memorized data in llms? a tale of two benchmarks. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3190–3211, 2024.

[12] Bowen Chen, Namgi Han, and Yusuke Miyao. A multi-perspective analysis of memorization in large language models. *arXiv preprint arXiv:2405.11577*, 2024.

[13] Ruizhe Chen, Jianfei Yang, Huimin Xiong, Jianhong Bai, Tianxiang Hu, Jin Hao, Yang Feng, Joey Tianyi Zhou, Jian Wu, and Zuozhu Liu. Fast model debias with machine unlearning. *Advances in Neural Information Processing Systems*, 36:14516–14539, 2023.

[14] Xiaoyu Cheng. Approaching memorization in large language models. 2025.

[15] Trung Cuong Dang and David Mohaisen. Memories retrieved from many paths: A multi-prefix framework for robust detection of training data leakage in large language models. *arXiv preprint arXiv:2511.20799*, 2025.

[16] Grégoire Delétang, Anian Ruoss, Paul-Ambroise Duquenne, Elliot Catt, Tim Genewein, Christopher Mattern, Jordi Grau-Moya, Li Kevin Wenliang, Matthew Aitchison, Laurent Orseau, et al. Language modeling is compression. In *International Conference on Learning Representations*, volume 2024, pages 14165–14181, 2024.

[17] Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. Enhancing chat language models by scaling high-quality instructional conversations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3029–3051, 2023.

[18] Vitaly Feldman and Chiyuan Zhang. What neural networks memorize and why: Discovering the long tail via influence estimation. *Advances in neural information processing systems*, 33:2881–2891, 2020.

[19] Yujie Feng, Li-Ming Zhan, Zexin Lu, Yongxin Xu, Xu Chu, Yasha Wang, Jiannong Cao, Philip S Yu, and Xiao-Ming Wu. Geoedit: Geometric knowledge editing for large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 13401–13416, 2025.

[20] Kunihiko Fukushima. Visual feature extraction by a multilayered network of analog threshold elements. *IEEE Transactions on Systems Science and Cybernetics*, 5(4):322–333, 2007.

[21] Chongyang Gao, Lixu Wang, Kaize Ding, Chenkai Weng, Xiao Wang, and Qi Zhu. On large language model continual unlearning. In *International Conference on Learning Representations*, volume 2025, pages 101772–101801, 2025.

[22] Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

[23] Laura Graves, Vineel Nagisetty, and Vijay Ganesh. Amnesiac machine learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 11516–11524, 2021.

[24] Mansi Gupta, Nikhar Waghela, Sarthak Gupta, Shourya Goel, and Sanjif Shanmugavelu. Pruning as a defense: Reducing memorization in large language models. *arXiv preprint arXiv:2502.15796*, 2025.

[25] Amr Hegazy, Mostafa Elhoushi, and Amr Alanwar. Guiding giants: Lightweight controllers for weighted activation steering in llms. *arXiv preprint arXiv:2505.20309*, 2025.

[26] Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*, 2018.

[27] Chenhui Hu, Pengfei Cao, Yubo Chen, Kang Liu, and Jun Zhao. Wilke: Wise-layer knowledge editor for lifelong knowledge editing. *arXiv preprint arXiv:2402.10987*, 2024.

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

[29] Jing Huang, Diyi Yang, and Christopher Potts. Demystifying verbatim memorization in large language models. *arXiv preprint arXiv:2407.17817*, 2024.

[30] Jiahao Huo, Yibo Yan, Xu Zheng, Yuanhuiyi Lyu, Xin Zou, Zhihua Wei, and Xuming Hu. Mmunlearner: Reformulating multimodal machine unlearning in the era of multimodal large language models. *arXiv preprint arXiv:2502.11051*, 2025.

[31] Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. *arXiv preprint arXiv:2212.04089*, 2022.

[32] Daphne Ippolito, Florian Tramèr, Milad Nasr, Chiyuan Zhang, Matthew Jagielski, Katherine Lee, Christopher A Choquette-Choo, and Nicholas Carlini. Preventing verbatim memorization in language models gives a false sense of privacy. *arXiv preprint arXiv:2210.17546*, 2022.

[33] Joel Jang, Dongkeun Yoon, Sohee Yang, Sungmin Cha, Moontae Lee, Lajanugen Logeswaran, and Minjoon Seo. Knowledge unlearning for mitigating privacy risks in language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14389–14408, 2023.

[34] Houcheng Jiang, Junfeng Fang, Ningyu Zhang, Guojun Ma, Mingyang Wan, Xiang Wang, Xiangnan He, and Tat-seng Chua. Anyedit: Edit any knowledge encoded in language models. *arXiv preprint arXiv:2502.05628*, 2025.

[35] Yuxin Jiang, Yufei Wang, Chuhan Wu, Wanjun Zhong, Xingshan Zeng, Jiahui Gao, Liangyou Li, Xin Jiang, Lifeng Shang, Ruiming Tang, et al. Learning to edit: Aligning llms with knowledge editing. *arXiv preprint arXiv:2402.11905*, 2024.

[36] Bruce W Lee, Inkit Padhi, Karthikeyan Natesan Ramamurthy, Erik Miehling, Pierre Dognin, Manish Nagireddy, and Amit Dhurandhar. Programming refusal with conditional activation steering. In *The Thirteenth International Conference on Learning Representations*, 2025.

[37] Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. Deduplicating training data makes language models better. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8424–8445, 2022.

[38] Xuan Liu, Haoyang Shang, and Haojian Jin. CoBRA: Programming cognitive bias in social agents using classic social science experiments. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, 2026.

[39] Samuel Marks and Max Tegmark. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. *arXiv preprint arXiv:2310.06824*, 2023.

[40] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. *Advances in neural information processing systems*, 35:17359–17372, 2022.

[41] Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. Mass-editing memory in a transformer. *arXiv preprint arXiv:2210.07229*, 2022.

[42] Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D Manning, and Chelsea Finn. Memory-based model editing at scale. In *International Conference on Machine Learning*, pages 15817–15831. PMLR, 2022.

[43] John X Morris, Chawin Sitawarin, Chuan Guo, Narine Kokhlikyan, G Edward Suh, Alexander M Rush, Kamalika Chaudhuri, and Saeed Mahloujifar. How much do language models memorize? *arXiv preprint arXiv:2505.24832*, 2025.

[44] Milad Nasr, Javier Rando, Nicholas Carlini, Jonathan Hayase, Matthew Jagielski, A Feder Cooper, Daphne Ippolito, Christopher A Choquette-Choo, Florian Tramèr, and Katherine Lee. Scalable extraction of training data from aligned, production language models. In *The Thirteenth International Conference on Learning Representations*, 2025.

[45] Thanh Tam Nguyen, Thanh Trung Huynh, Zhao Ren, Phi Le Nguyen, Alan Wee-Chung Liew, Hongzhi Yin, and Quoc Viet Hung Nguyen. A survey of machine unlearning. *ACM Transactions on Intelligent Systems and Technology*, 16(5):1–46, 2025.

[46] Chenjie Ni, Zhepeng Wang, Runxue Bao, Shangqian Gao, and Yanfu Zhang. Controllable memorization in llms via weight pruning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 15141–15156, 2025.

[47] Yifu Qiu, Zheng Zhao, Yftah Ziser, Anna Korhonen, Edoardo M Ponti, and Shay B Cohen. Spectral editing of activations for large language model alignment. *Advances in Neural Information Processing Systems*, 37:56958–56987, 2024.

[48] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[49] Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. Steering llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15504–15522, 2024.

[50] Elena Sofia Ruzzetti, Giancarlo A Xompero, Davide Venditti, and Fabio Massimo Zanzotto. Private memorization editing: Turning memorization into a defense to strengthen data privacy in large language models. *arXiv preprint arXiv:2506.10024*, 2025.

[51] Mansi Sakarvadia, Aswathy Ajith, Arham Khan, Nathaniel Hudson, Caleb Geniesse, Kyle Chard, Yaoqing Yang, Ian Foster, and Michael W Mahoney. Mitigating memorization in language models. *arXiv preprint arXiv:2410.02159*, 2024.

[52] Ali Satvaty, Anna Visman, Dan Seidel, Suzan Verberne, and Fatih Turkmen. Memorization is language-sensitive: Analyzing memorization and inference risks of llms in a multilingual setting. In *Proceedings of the First Workshop on Large Language Model Memorization (L2M2)*, pages 106–126, 2025.

[53] William F Shen, Xinchi Qiu, Meghdad Kurmanji, Alex Iacob, Lorenzo Sani, Yihong Chen, Nicola Cancedda, and Nicholas D Lane. Lunar: Llm unlearning via neural activation redirection. *Thirty-nineth Conference on Neural Information Processing Systems*, 2025.

[54] Leheng Sheng, Changshuo Shen, Weixiang Zhao, Junfeng Fang, Xiaohao Liu, Zhenkai Liang, Xiang Wang, An Zhang, and Tat-Seng Chua. Alphasteer: Learning refusal steering with principled null-space constraint. *arXiv preprint arXiv:2506.07022*, 2025.

[55] Dan Shi, Tianhao Shen, Yufei Huang, Zhigen Li, Yongqi Leng, Renren Jin, Chuang Liu, Xinwei Wu, Zishan Guo, Linhao Yu, et al. Large language model safety: A holistic survey. *arXiv preprint arXiv:2412.17686*, 2024.

[56] Niklas Stoehr, Kevin Du, Vésteinn Snæbjarnarson, Robert West, Ryan Cotterell, and Aaron Schein. Activation scaling for steering and interpreting language models. *arXiv preprint arXiv:2410.04962*, 2024.

[57] Manan Suri, Nishit Anand, and Amisha Bhaskar. Mitigating memorization in llms using activation steering. *arXiv preprint arXiv:2503.06040*, 2025.

[58] Ayush K Tarun, Vikram S Chundawat, Murari Mandal, and Mohan Kankanhalli. Fast yet effective machine unlearning. *IEEE Transactions on Neural Networks and Learning Systems*, 35(9):13046–13055, 2023.

[59] Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J Vazquez, Ulisse Mini, and Monte MacDiarmid. Steering language models with activation engineering. *arXiv preprint arXiv:2308.10248*, 2023.

[60] Han Wang, Gang Wang, and Huan Zhang. Steering away from harm: An adaptive approach to defending vision language model against jailbreaks. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 29947–29957, 2025.

[61] Lingzhi Wang, Xingshan Zeng, Jinsong Guo, Kam-Fai Wong, and Georg Gottlob. Selective forgetting: Advancing machine unlearning techniques and evaluation in language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 843–851, 2025.

[62] Peng Wang, Ningyu Zhang, Bozhong Tian, Zekun Xi, Yunzhi Yao, Ziwen Xu, Mengru Wang, Shengyu Mao, Xiaohan Wang, Siyuan Cheng, et al. Easyedit: An easy-to-use knowledge editing framework for large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 82–93, 2024.

[63] Xiuyuan Wang, Chaochao Chen, Weiming Liu, Xinting Liao, Fan Wang, and Xiaolin Zheng. Efficient source-free unlearning via energy-guided data synthesis and discrimination-aware multitask optimization. In *Forty-second International Conference on Machine Learning*, 2025.

[64] Jiaxin Wen, Pei Ke, Hao Sun, Zhexin Zhang, Chengfei Li, Jinfeng Bai, and Minlie Huang. Unveiling the implicit toxicity in large language models. *arXiv preprint arXiv:2311.17391*, 2023.

[65] Yuxin Wen, Yangsibo Huang, Tom Goldstein, Ravi Kumar, Badih Ghazi, and Chiyuan Zhang. Quantifying cross-modality memorization in vision-language models. *arXiv preprint arXiv:2506.05198*, 2025.

[66] Zhengxuan Wu, Aryaman Arora, Atticus Geiger, Zheng Wang, Jing Huang, Dan Jurafsky, Christopher D Manning, and Christopher Potts. Axbench: Steering llms? even simple baselines outperform sparse autoencoders. *arXiv preprint arXiv:2501.17148*, 2025.

[67] Chulin Xie, Yangsibo Huang, Chiyuan Zhang, Da Yu, Xinyun Chen, Bill Yuchen Lin, Bo Li, Badih Ghazi, and Ravi Kumar. On memorization of large language models in logical reasoning. In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pages 2742–2785, 2025.

[68] Alexander Xiong, Xuandong Zhao, Aneesh Pappu, and Dawn Song. The landscape of memorization in llms: Mechanisms, measurement, and mitigation. *arXiv preprint arXiv:2507.05578*, 2025.

[69] Xiaoyu Xu, Xiang Yue, Yang Liu, Qingqing Ye, Huadi Zheng, Peizhao Hu, Minxin Du, and Haibo Hu. Unlearning isn't deletion: Investigating reversibility of machine unlearning in llms. *arXiv preprint arXiv:2505.16831*, 2025.

[70] Jin Yao, Eli Chien, Minxin Du, Xinyao Niu, Tianhao Wang, Zezhou Cheng, and Xiang Yue. Machine unlearning of pre-trained large language models. *arXiv preprint arXiv:2402.15159*, 2024.

[71] Shenglai Zeng, Yaxin Li, Jie Ren, Yiding Liu, Han Xu, Pengfei He, Yue Xing, Shuaiqiang Wang, Jiliang Tang, and Dawei Yin. Exploring memorization in fine-tuned language models. *arXiv preprint arXiv:2310.06714*, 2023.

[72] Ruiqi Zhang, Licong Lin, Yu Bai, and Song Mei. Negative preference optimization: From catastrophic collapse to effective unlearning. *arXiv preprint arXiv:2404.05868*, 2024.

[73] Terry Jingchen Zhang, Gopal Dev, Ning Wang, Nicole Ni, Wenyuan Jiang, Yinya Huang, Bernhard Schölkopf, Mrinmaya Sachan, and Zhijing Jin. Beyond memorization: Reasoning-driven synthesis as a mitigation strategy against benchmark contamination. In *First Workshop on Foundations of Reasoning in Language Models*, 2025.

[74] Zhaohan Zhang, Ziquan Liu, and Ioannis Patras. Get confused cautiously: Textual sequence memorization erasure with selective entropy maximization. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 10924–10939, 2025.

[75] Weixiang Zhao, Jiahe Guo, Yulin Hu, Yang Deng, An Zhang, Xingyu Sui, Xinyang Han, Yanyan Zhao, Bing Qin, Tat-Seng Chua, et al. Adasteer: Your aligned llm is inherently an adaptive jailbreak defender. *arXiv preprint arXiv:2504.09466*, 2025.

Appendix contents

A Theoretical foundations and scope **17**

A.1 Excess code length and token-wise decomposition 17

A.2 Validity and limitations of the signal 17

A.3 Calibration gradients and accumulation 18

A.4 Proof of the whitened rank-K spectral optimum 18

A.5 A fixed-direction gate perturbation bound 19

A.6 Second-moment normalization and thresholds 19

A.7 Correction coefficients 19

B Experimental protocols and metrics **19**

B.1 Evaluation data and metrics 20

B.2 Baseline mechanisms and calibration access 20

B.3 Uncertainty and observed zeros 20

C Signal validation and reference dependence **21**

C.1 Concentration of positive likelihood mass 21

C.2 Exposure validation versus verbatim ranking 21

C.3 Reference exposure controls 21

D Gate configuration, selectivity, and utility **23**

D.1 Pythia-2.8B: consecutive gate comparison 23

D.2 Qwen2.5-7B: gate transfer and strength sweep 23

D.3 Run requirement on memorized and legitimate cohorts 24

D.4 Token classes, mask controls, and causal tests 25

D.5 Threshold calibration under distribution shift 26

D.6 Gated versus ungated results 27

E Additional models, domains, and utility measures **27**

E.1 Qwen2.5-7B accuracy and judge evaluation 27

E.2 Real-document domains 27

E.3 Synthetic PII: duplication and extraction templates 28

E.4 Approximate and semantic reproduction 30

E.5 Extraction attacks 31

F Further ablations and additional results **31**

F.1 Probe discrimination and stability 31

F.2 Long-form generation 31

F.3 Layer selection 31

F.4 Bootstrap stability of activation subspaces 32

F.5 Utility-constrained modern-model results 32
F.6 Qualitative examples 32

G Methodological connections **34**
G.1 Low-rank updates 34
G.2 Relation to discriminant analysis 34
G.3 Unlearning and editing 34

H Broader impact and limitations **35**
H.1 Societal implications 35
H.2 Limitations 35
H.3 Future directions 35

A Theoretical foundations and scope

This appendix collects the derivations behind GSS, the experimental protocols, and additional results. Throughout, u denotes a probe direction and v a steer direction, following the main text.

A.1 Excess code length and token-wise decomposition

For fixed target and reference distributions, the model-based code lengths of $x_{1:T}$ are $-\log p_\theta(x_{1:T})$ and $-\log p_{\mathrm{ref}}(x_{1:T})$. This interpretation follows the prediction–compression connection studied by Delétang et al. [16]. Their difference is

$$S(x) = \log p_\theta(x) - \log p_{\mathrm{ref}}(x) = \sum_{t=1}^{T} [\log p_\theta(x_t \mid x_{<t}) - \log p_{\mathrm{ref}}(x_t \mid x_{<t})] = \sum_t \omega_t. \quad (9)$$

This decomposition is exact for autoregressive models evaluated on the same sequence and compatible tokenization. Models with different tokenizers require an explicitly aligned scoring protocol; token-wise differences are not directly comparable otherwise. Logarithms are natural unless stated otherwise.

For any data distribution q for which the quantities exist,

$$\mathbb{E}_q[-\log p(x)] = H(q) + \mathrm{KL}(q\|p), \qquad \mathbb{E}_q[S(x)] = \mathrm{KL}(q\|p_{\mathrm{ref}}) - \mathrm{KL}(q\|p_\theta). \quad (10)$$

Positive excess likelihood can therefore reflect a better fit to reusable structure, sequence-specific exposure, or both; it is not by itself a measure of unintended memorization. Its usefulness depends on a meaningful reference, which we assess empirically in Appendix C.3.

The information-theoretic motivation does not turn this likelihood difference into an exact mutual-information estimator. In particular, the conditional mutual-information identity is

$$I(X; \Theta \mid \Theta_{\mathrm{ref}}) = H(X \mid \Theta_{\mathrm{ref}}) - H(X \mid \Theta, \Theta_{\mathrm{ref}}),$$

which generally cannot drop Θ_{ref} from the second term. Likewise, model code length is not an exact evaluation of Kolmogorov complexity. We use excess likelihood as an operational signal rather than asserting either of these equalities for a particular pair of trained models.

Positive mass is a different quantity. Let $m_t = [\omega_t]_+$ and $\mathcal{M}_+(x) = \sum_t m_t$. Then $[S(x)]_+ \leq \mathcal{M}_+(x)$, with inequality possible because negative contributions cancel positive ones in $S(x)$ but not in $\mathcal{M}_+$. The concentration analysis concerns $\mathcal{M}_+$; sequence-level exact reproduction remains the mitigation outcome.

A.2 Validity and limitations of the signal

A domain-competent reference with lower exposure helps separate exposure-related predictive gains from predictability shared by both models; it need not be more accurate than the target. High-signal

positions are therefore candidates rather than certified memorized tokens, and low-signal positions are not certified generalized tokens. Task learning, forgetting, model capacity, optimization noise, and reference exposure can all affect the score. Appendix C.2 reports exposure controls.

Whitening normalizes a second moment but does not remove false positives. Singular-vector perturbation bounds would require explicit perturbation-norm and spectral-gap conditions, so we do not rely on them for robustness to the choice of reference.

Relation to membership and counterfactual analyses. Likelihood-ratio tests also arise in membership inference, where inclusion and exclusion models calibrate example-specific prediction difficulty [6]. GSS does not implement that membership test: its target–reference token score is not a calibrated membership probability. Likewise, inclusion-controlled validation is motivated by counterfactual approaches to memorization [18], but exposure sensitivity alone does not establish whether a particular prediction is harmful or whether it will lead to verbatim generation.

A.3 Calibration gradients and accumulation

For a batch $\mathcal{B}$, let w_t^a be the fixed high- or low-signal mask, including valid prediction positions, and $n_a = \sum_{\mathcal{B}} w_t^a$. For $n_a > 0$, we differentiate

$$\mathcal{L}_a = -n_a^{-1} \sum_{\mathcal{B}} w_t^a \log p_\theta(x_t \mid x_{<t}), \qquad g_t^a = \nabla_{h_t} \mathcal{L}_a.$$

The target weights are not updated. The reference and masks are not backpropagated through. The main-text gradient $g_t = \nabla_{h_t} \ell_{\mathcal{H}}$ is the high-signal log-likelihood gradient, so $g_t = -g_t^{\mathrm{hi}}$ up to this normalization, and M_{hi} in Section 4 corresponds to $-\sum_t h_t (g_t^{\mathrm{hi}})^\top$. At fixed positive mask, the CE gradient is the negative of the positive likelihood-gap gradient, up to normalization. Consequently, $\Delta\mathcal{L}_a \simeq (g_t^a)^\top \Delta h_t$ describes an increase in CE only when this inner product is positive. A positive singular-value objective alone does not establish that property for a subtractive update.

Gradients are normalized within each batch or document, so the accumulated matrices are sums of these normalized terms rather than uniform expectations over positions. Gradients at a position also include downstream causal dependencies and are therefore not, in general, gradients of that position's next-token loss alone.

Contrastive constructions. The contrastive configurations accumulate $M^a = \sum_t c_t^a \, h_t (g_t^a)^\top$ for $a \in \{\mathrm{hi}, \mathrm{lo}\}$ over the calibration sequences and set $M_{\mathrm{diff}} = M^{\mathrm{hi}} - M^{\mathrm{lo}}$, with the CE gradients g_t^a defined above and no whitening. The two configurations differ in the accumulation weight c_t^a. C sets $c_t^a = 1$ at every valid position, so the high- or low-signal mask enters only through the loss $\mathcal{L}_a$. C-M sets $c_t^a = w_t^a$ and accumulates only at the masked positions. Its masks cover the answer tokens of the fine-tuning items for Qwen2.5-7B on GSM8K and synthetic PII, and every token of the memorized and control documents for the real-document domains. The probe is the leading right singular vector of M_{diff} and the steer the leading left singular vector (Appendix A.4). In the experiments of Appendices D and E.1, both vectors are negated together when needed, so that the mean activation on high-signal positions minus the mean activation on low-signal positions has a positive probe response.

A.4 Proof of the whitened rank-K spectral optimum

Let $C = \Sigma_{\mathrm{lo}} + \gamma I = LL^\top \succ 0$ and $A = L^{-1} M_{\mathrm{hi}}$. The spectral problem of the main text is

$$\max_{U,V} \operatorname{tr}(U^\top M_{\mathrm{hi}} V) \quad \text{s.t.} \quad U^\top C U = I_K, \quad V^\top V = I_K. \tag{11}$$

Set $Q = L^\top U$. This maps the constraints to $Q^\top Q = I_K$ and $V^\top V = I_K$, and the objective to $\operatorname{tr}(Q^\top A V)$. For any feasible V, maximizing over Q gives the nuclear norm $\|AV\|_*$. The K singular values of AV are bounded by the corresponding leading singular values of A, so

$$\operatorname{tr}(Q^\top A V) \le \|AV\|_* \le \sum_{k=1}^{K} \sigma_k(A).$$

If $A = \widetilde{U}S\widetilde{V}^\top$, choosing $Q = \widetilde{U}_{:,1:K}$ and $V = \widetilde{V}_{:,1:K}$ attains equality. Hence $U = L^{-\top}\widetilde{U}_{:,1:K}$ is optimal, proving Theorem 4.1. Degenerate singular values can yield non-unique optimal bases. The result concerns the spectral objective only.

For contrastive $M_{\text{diff}} = PSQ^\top$, the same variational argument maximizes $\operatorname{tr}(V^\top M_{\text{diff}} U)$ at $V = P_{:,1:K}$, $U = Q_{:,1:K}$ under Euclidean orthogonality. Here the left and right singular vectors serve as steer and probe, respectively; this construction is not equivalent to the whitened one.

A.5 A fixed-direction gate perturbation bound

For fixed directions u, v, a random activation h, and a gradient g, define $J_0 = \mathbb{E}[(u^\top h)(g^\top v)]$ and $J_\epsilon = \mathbb{E}[\mathbb{I}(|u^\top h| > \epsilon)(u^\top h)(g^\top v)]$. If $\mathbb{E}|g^\top v| < \infty$, then

$$|J_0 - J_\epsilon| \le \epsilon \mathbb{E}[\mathbb{I}(|u^\top h| \le \epsilon)|g^\top v|] \le \epsilon \mathbb{E}|g^\top v|. \tag{12}$$

Indeed, the difference integrates only over $|u^\top h| \le \epsilon$, where the activation factor is bounded by ϵ. The bound concerns fixed directions only. It does not compare against an optimum over gated directions, give a guarantee for nonlinear generation, or cover the one-sided excess and consecutive-position gates.

A.6 Second-moment normalization and thresholds

The whitened configuration estimates the uncentered second moment $\Sigma_{\text{lo}} = N_{\text{lo}}^{-1} \sum h_t h_t^\top$; no calibration mean is subtracted during inference. For $\epsilon > 0$, Markov's inequality applied to $(u^\top h)^2$ gives

$$\Pr(|u^\top h| > \epsilon) \le \frac{u^\top \Sigma_{\text{lo}} u}{\epsilon^2}.$$

This bound holds for the calibration distribution and does not by itself guarantee utility preservation. With jitter, $L^{-1}\Sigma_{\text{lo}}L^{-\top}$ need not equal I exactly, and whitening does not guarantee linear separation of high- and low-signal positions. A 90th-percentile threshold places about 10% of calibration responses above the threshold. With multiple probes, the per-probe exceedance rate differs from the overall rate of intervention.

A.7 Correction coefficients

For whitened directions, the update is $h' = h - 0.1\lambda \sum_k \mathbb{I}(|u_k^\top h| > \epsilon)(u_k^\top h)v_k$, so the strength in Eq. (8) is $\beta = 0.1\lambda$. For a rank-one contrastive direction,

$$h' = h - \beta \frac{[u^\top h - \epsilon]_+}{\max(u^\top v, 10^{-3})} v.$$

Here $[z]_+ = \max(z, 0)$ is the rectified linear function [20]. This one-sided, threshold-subtracted update is not an instance of Eq. (8), which applies to the whitened configuration; for C and C-M, β denotes the scale of this update. The base contrastive configuration uses a fixed scale of 0.2; the other configurations set the scale explicitly. If $u^\top v \ge 10^{-3}$, $\beta = 1$, and one direction is active, the corrected probe response equals ϵ. With a clipped or negative inner product, this conclusion is not valid. In particular, clipping a negative inner product to a positive floor does not preserve its sign. For multiple directions, $u_i^\top h' = u_i^\top h - \sum_k a_k u_i^\top v_k$ includes cross terms, so the update does not in general project jointly onto all threshold boundaries.

B Experimental protocols and metrics

Notation. We write W for the whitened configuration, C for the contrastive configuration, and C-M for its variant with position-masked accumulation, used in all Qwen2.5-7B experiments (Appendix A.3). The threshold ϵ is set to the $(1 - \eta)$ quantile of absolute probe responses for W and signed probe responses for C/C-M on calibration text, so η is the nominal per-probe exceedance rate; r is the consecutive-position requirement of the gate, and $r = 1$ denotes the single-position gate.

Gate configurations. Tables 1 and 2 and Figures 4–7 in Section 5 use the single-position gate ($r = 1$). Table 1 uses the whitened configuration (W) and Table 2 the base contrastive configuration (C), both with rank one. In the appendix, every result uses $r = 1$ unless its table lists r. Table 4 lists the tables that report the two-consecutive-position gate ($r = 2$), and Table 3 in Section 5.6 collects one operating point from each setting. In each of them, $r = 2$ is compared with $r = 1$ under the same model, directions, threshold, and strength; the $r = 2$ rows at $\beta \in \{20, 30\}$ in Table 12 are additional strengths without an $r = 1$ counterpart.

Table 4: Gate configurations of the results that report $r = 2$. All directions have rank one, and ϵ is the $(1 - \eta)$ quantile of probe responses on calibration text.

Table	Model	Data	Directions	Layer	η	β	r
10	Pythia-2.8B	Pile	C	22	0.1	3, 10	1, 2
13	Pythia-2.8B	Pile	C	22	0.1	10	1, 2, 3
11	Qwen2.5-7B	GSM8K	C-M	21	0.1	3	1, 2
12	Qwen2.5-7B	GSM8K	C-M	21	0.1	10, 20, 30	1, 2
14	Qwen2.5-7B	GSM8K	C-M	21	0.1	3	1, 2, 3
21	Qwen2.5-7B	Real documents	C-M	21	0.05, 0.1	1, 3, 10	1, 2
22	Qwen2.5-7B	Synthetic PII	C-M	21	0.1	3, 10	1, 2

B.1 Evaluation data and metrics

On Pythia, we use $n = 72$, $k = 32$, and a 40-token suffix. The main experiments use a pool of 1,632 sequences from The Pile [22]; several additional experiments use a separate pool of 505 extracted sequences together with random Pile controls. Because the two populations differ, we do not compare baseline rates across them.

On Qwen2.5-7B, we measure exact reproduction of the first 30 response tokens on exposed training items and final-answer accuracy on 300 test items, generating up to 160 tokens. The real-document experiments measure exact reproduction of the first 50 continuation tokens after a 50-token prefix. We report these outcomes under their own names rather than as a single (n, k) metric, and we distinguish teacher-forced token matches from matches in free-running generation.

B.2 Baseline mechanisms and calibration access

During calibration, GSS requires a reference model, target gradients, and calibration examples, but it updates no parameters. Table 5 summarizes the requirements of each baseline. The LUNAR implementation we evaluate optimizes a down-projection matrix, so we classify it as a weight-updating method.

Table 5: Weight updates and data access required by each method as evaluated.

Method	Target weight update	Calibration / optimization access
GSS	No	Target, reference, calibration corpus
NPO implementation	Yes	Forget loss, frozen reference scores, retain loss
Task Vector	Yes	Fine-tuning corpus and parameter copies
LUNAR implementation	Yes	Unlearning/reference activations
ROME / MEMIT	Yes	Edit requests and implementation-specific statistics

For NPO, we use AdamW with learning rate 10^{-5}, batch size 16, preference parameter 0.1, retain weight 50, and up to five epochs, with early stopping when retain loss exceeds three times its initial value. For Task Vector, we use AdamW with learning rate 10^{-2}, batch size 32, and ten epochs.

B.3 Uncertainty and observed zeros

We use sequence-level bootstrap resampling because positions within a sequence are dependent. Seed ranges and standard deviations describe variation across seeds and are not confidence intervals. For $z = 0$ observed successes among N independent Bernoulli trials, the two-sided 95% exact binomial upper limit is $1 - 0.025^{1/N}$; the one-sided 95% upper limit is $1 - 0.05^{1/N}$. For $N = 998$, these

are approximately 0.369% and 0.300%, respectively. A bootstrap over an all-zero vector yields a degenerate interval and does not establish zero population risk. Because repeated attacks on the same sequence are correlated, we summarize attack success per sequence at the stated budget.

C Signal validation and reference dependence

C.1 Concentration of positive likelihood mass

For nonnegative masses $m_i = [\omega_i]_+$ with total $S_+ > 0$, define the share carried by the largest 10% of positions, the Gini coefficient $\sum_{i,j} |m_i - m_j|/(2NS_+)$, and normalized effective support $S_+^2/(N \sum_i m_i^2)$. Zero-mass positions remain in the denominator. For $S_+ = 0$, concentration is undefined rather than evidence of sparsity. Positive-position frequency is reported separately from mass concentration.

Table 6: Concentration of positive likelihood mass: ranges across twelve settings (11,833 sequences, 1.72 million positions).

Statistic	Range
Top-10% positive-mass share	30–89%
Relative to uniform allocation	3.0–8.9×
Gini coefficient	0.50–0.91
Normalized effective support	0.088–0.532
Positions with $\omega > 0$	47–94%

These statistics indicate nonuniform positive mass rather than a specific parametric heavy tail: exponential tails fit better than power-law tails in nine of twelve settings. Theorem 4.1 assumes neither the tail shape nor concentration.

On Qwen3–GSM8K, the signal has skewness 2.89, a mean positive-run length of 1.64, and 60.8% singleton runs. These describe runs of positive signal, not lengths of memorized spans.

C.2 Exposure validation versus verbatim ranking

In seed-controlled canary experiments, ω identifies the top decile of inclusion gain with AUROC 0.99, and Pythia-2.8B scores reach 0.95 against independent Pythia-6.9B labels; the corresponding argmax-flip AUROCs are 0.83 and 0.81. On a separate pool of 505 extracted and 1,024 control sequences, positive-ω positions are about 7.67 times as likely as nonpositive ones to begin a 20-token verbatim continuation.

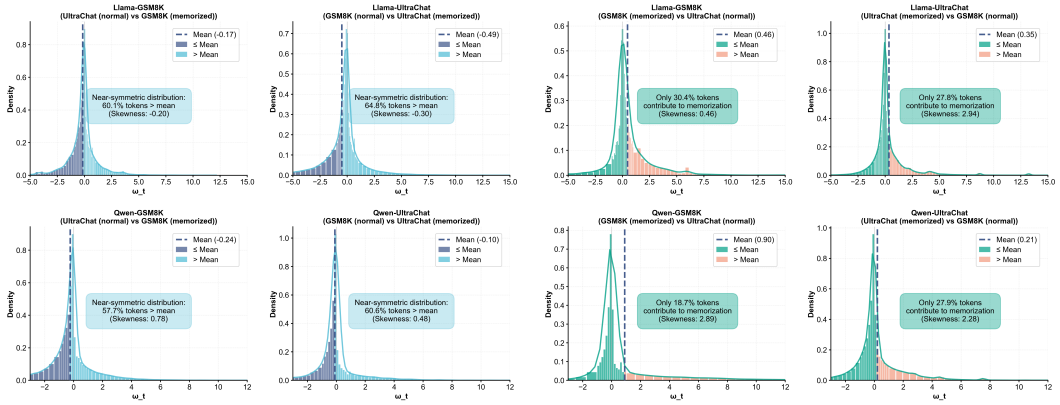
Partition ablation. On TinyMem, we replace the high-signal calibration subset with a random or an inverse subset. This tests how useful the selected subset is for constructing an intervention, not whether it identifies memorized tokens.

C.3 Reference exposure controls

We separate two questions: whether the reference provides an exposure-sensitive signal, and whether the calibrated intervention remains effective across a particular set of reference checkpoints. Similar post-intervention reproduction rates do not, by themselves, establish that the signal measures exposure.

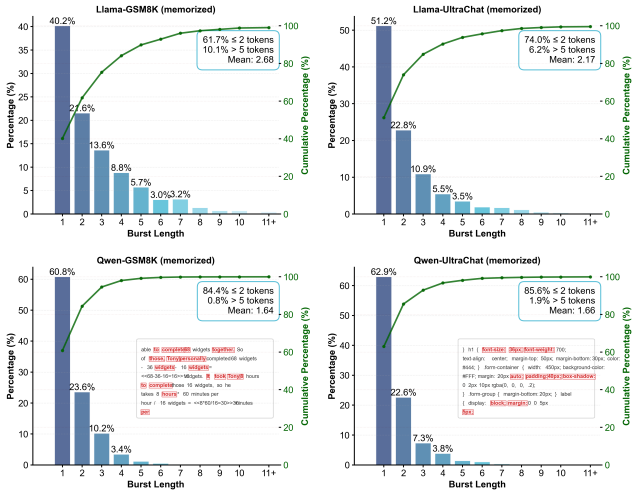
References without the original checkpoint. The reference need not be the target's original or immediately preceding checkpoint. A retained pre-exposure checkpoint, an independently trained model, or a domain-competent model trained on disjoint data can provide a useful comparison when it has lower exposure to the evaluated sequences. Compatible tokenization is required for the token-wise comparison used here.

Table 8 compares references with exposure contrast against a perturbed copy of the target. The weaker performance of the clone supports the role of exposure contrast in this control; disagreement introduced by perturbation is not sufficient to recover the same discrimination. An observed AUC of 1.00 does not guarantee perfect discrimination on new data.



(a) Signal distributions on control data.

(b) Signal distributions on reproduced sequences.



(c) Run lengths of positive signal.

Figure 8: Distribution of the likelihood signal ω_t. Low signal is not a generalization label, distributional shape is setting-dependent, and run frequency does not measure intervention magnitude or causal importance.

Table 7: TinyMem partition ablation. Calibrating on high-signal positions gives the lowest reproduction.

Partition	Math Noise			Math Backdoor			LM Noise			LM Backdoor		
	Mem. ↓	Acc. ↑	Time ↓	Mem. ↓	Acc. ↑	Time ↓	Mem. ↓	PPL ↓	Time ↓	Mem. ↓	PPL ↓	Time ↓
Random partition	18.42	96.71	0.001	21.36	96.44	0.001	27.85	64.02	0.002	31.17	64.36	0.003
Inverse partition ($\omega_t < 0$)	34.79	96.55	0.001	39.28	96.21	0.001	46.63	64.71	0.002	49.85	65.08	0.003
High-ω_t partition (GSS)	**0.00**	**96.98**	0.001	**0.00**	**96.82**	0.001	**0.00**	**63.13**	0.002	**0.00**	**63.17**	0.003

Exposure dose response. As the reference becomes increasingly exposed to the evaluated sequences, verbatim-20 enrichment decreases from approximately $36.9\times$ to $1.4\times$, and sequence-level AUC decreases from 0.92 to 0.49. This diagnostic is distinct from the post-intervention sweep in Table 9. Enrichment near one indicates little localization advantage under that reference and evaluation protocol. Thus, lower exposure is an empirical requirement for the intended interpretation of the signal, rather than robustness to arbitrary references.

In an exploratory control of predictable but unexposed text, the false-positive rate at 90% recall is 16.5% for target confidence and 0% for ω. Unlike target confidence, ω thus separates exposure from mere predictability in this experiment.

Table 8: Controlled reference comparison: sequence-level discrimination. These finite-sample results test exposure contrast rather than universal reference interchangeability. No intervention gate is applied in this diagnostic.

Reference	Exposure contrast	AUC
Matched unexposed model	Yes	1.00
Independent-seed unexposed model	Yes	1.00
Model trained on disjoint data	Yes	1.00
Noise-perturbed target clone	No	0.58

Post-intervention checkpoint sensitivity. Table 9 reports reproduction and PPL after intervention as the reference checkpoint varies.

Table 9: Pythia-2.8B reproduction and PPL across reference checkpoints ($r = 1$).

Reference Model Step	Mem. Rate	PPL
2000	6.45%	31.49
4000	7.11%	29.49
6000	6.93%	28.26
8000	7.11%	28.64
10000	7.11%	28.74

D Gate configuration, selectivity, and utility

All consecutive-gate ($r = 2$) experiments in this section use contrastive configurations, C on Pythia-2.8B and C-M on Qwen2.5-7B. They characterize the run requirement under these configurations and do not directly validate the whitened configuration analyzed in Section 4.

D.1 Pythia-2.8B: consecutive gate comparison

The target is Pythia-2.8B-deduped at step 143000 and the reference at step 2000. We estimate rank-one C directions at layer 22 (zero-indexed) on 70% of the extracted pool and evaluate on the remaining 30%, for split seeds 42, 43, and 44. The threshold is the 90th percentile of the signed projection on the last 1,024 random Pile calibration sequences, and PPL uses the first 800. The table reports means over the three splits; the target model is not retrained.

Table 10: Pythia-2.8B, rank-one C, $\eta = 0.1$. Trigger is the fraction of evaluated positions at which the gate fires; number preservation measures overlap with numbers in the reference continuation and is not task accuracy.

β	r	Exact	PPL	Trigger	Number preservation
3	1	0.1667	23.45	0.435	0.434
3	2	0.1732	23.15	0.325	0.439
10	1	0.0965	25.04	0.379	0.333
10	2	0.1118	24.15	0.271	0.379

At $\beta = 10$, requiring two consecutive exceedances lowers the trigger rate and PPL but raises residual exact reproduction from 0.0965 to 0.1118, so the run requirement trades suppression for utility. Residual rates reported elsewhere in this appendix (e.g., Table 15) use a different pool and protocol and are not directly comparable.

D.2 Qwen2.5-7B: gate transfer and strength sweep

On Qwen2.5-7B, we compare the two gates at matched strength on GSM8K with rank-one C-M directions, $\eta = 0.1$, and 30-token reproduction, using a single seed. Reproduction is exact match of the first 30 response tokens on exposed training items, and utility is final-answer accuracy on 300 GSM8K test items (Appendix B.1).

Table 11: Qwen2.5-7B on GSM8K, C-M, $\beta = 3$, $\eta = 0.1$, single seed. Accuracy is measured on 300 test items.

r	Exact-30	Accuracy	Trigger	Number preservation
1	0.515	0.533	0.078	0.854
2	0.770	0.600	0.021	0.937

The consecutive gate intervenes less often and preserves more of the output, but substantially weakens suppression. Table 12 reports a three-seed strength sweep. The sweep reports number preservation rather than test accuracy, and number preservation is not a substitute for test accuracy.

Table 12: Qwen2.5-7B gate strength sweep with C-M, $\eta = 0.1$, and 30-token exact reproduction, reported per seed.

Seed	r	β	Exact-30	Number preservation	Trigger
1	1	10	0.243	0.697	0.076
1	2	10	0.622	0.863	0.021
1	2	20	0.566	0.836	0.020
1	2	30	0.536	0.814	0.018
2	1	10	0.137	0.533	0.173
2	2	10	0.387	0.669	0.100
2	2	20	0.300	0.596	0.099
2	2	30	0.271	0.571	0.099
44	1	10	0.333	0.756	0.078
44	2	10	0.654	0.889	0.021
44	2	20	0.581	0.853	0.021
44	2	30	0.536	0.833	0.022

Increasing strength does not make the two-position gate uniformly better, so the run requirement should be chosen for each model and data setting.

D.3 Run requirement on memorized and legitimate cohorts

Tables 13 and 14 vary the run requirement over $r \in \{1, 2, 3\}$ and separate the effect of the gate on memorized sequences from its effect on legitimate ones. Both cohorts are fixed at baseline, before any intervention. Trigger is the share of sequences in which the gate fires at least once during generation, and selectivity is the memorized trigger divided by the legitimate trigger. Each table uses a single estimate of the directions.

For Pythia-2.8B, the target, reference, and layer 22 are those of Table 10. The rank-one sign-fixed C direction is estimated once on the full 505-sequence pool, so the memorized cohort lies inside the estimation pool. The strength is $\beta = 10$, and ε is the 90th percentile of the projection on the same Pile calibration rows as in Table 10. The memorized cohort contains the 480 pool sequences with a teacher-forced 20-token verbatim run, and the legitimate cohort contains 1,174 random Pile sequences with no such run. Decoding is greedy for 40 tokens after a 32-token prompt, and Verbatim-20 is exact reproduction of the first 20 generated tokens. On the legitimate cohort, correct→incorrect is the share of teacher-forced tokens (positions 16 onward) that the target predicts correctly without the gate and incorrectly with it, and ΔAcc is the paired change in teacher-forced token accuracy.

Table 13: Pythia-2.8B cohorts, rank-one C, $\beta = 10$, $\eta = 0.1$. Values in %, except selectivity and PPL. Verbatim-20 is 66.5 without the gate, and PPL on the first 800 random Pile sequences is 22.35.

	Memorized ($n = 480$)			Legitimate ($n = 1{,}174$)				
r	Trigger	Verbatim-20	Reduction	Trigger	Corr.→incorr.	ΔAcc	Selectivity	PPL
1	96.9	14.2	52.3	78.4	5.8	-1.6	1.23	24.90
2	89.4	17.5	49.0	49.7	3.3	-1.0	1.80	24.03
3	85.2	22.1	44.4	31.5	2.0	-0.6	2.70	23.53

For Qwen2.5-7B, the adapter, direction, layer 21, and $\beta = 3$ are those of Table 11. The threshold ε is the 90th percentile of the projection on GSM8K test items 300–699, which are disjoint from the evaluated test items. The memorized cohort contains the 884 of the 1,000 SFT items whose first 30 answer tokens are reproduced at baseline, and the legitimate cohort contains the 166 of the first 300 test items that are answered correctly at baseline and share no 20-token span with any SFT answer. Decoding is greedy for 160 tokens. Correct→incorrect is the share of baseline-correct legitimate items that become incorrect, and ΔAcc is the change in final-answer accuracy on the legitimate cohort.

Table 14: Qwen2.5-7B on GSM8K cohorts, rank-one C-M, $\beta = 3$, $\eta = 0.1$, single seed. Values in %, except selectivity. Without the gate, the memorized cohort has Verbatim-30 99.7 and the legitimate cohort has accuracy 98.2.

	Memorized ($n = 884$)			Legitimate ($n = 166$)			
r	Trigger	Verbatim-30	Reduction	Trigger	Corr.→incorr.	ΔAcc	Selectivity
1	100.0	53.8	45.8	100.0	21.5	−20.5	1.00
2	89.3	84.8	14.8	83.7	4.9	−4.2	1.07
3	58.0	95.2	4.4	48.8	1.8	−1.8	1.19

On Pythia-2.8B, raising r lowers the legitimate trigger rate faster than the memorized one, and $r = 2$ keeps 49.0 of the 52.3 points of verbatim reduction. On Qwen2.5-7B, the two cohorts trigger at similar rates for every r, and $r = 2$ keeps 14.8 of the 45.8 points. These tables differ from Tables 10 and 11 in four respects: cohorts selected at baseline instead of a held-out split (Pythia) or all training items (Qwen), sequence-level instead of token-level trigger rates, verbatim reproduction of the first 20 tokens instead of exact match of all 40 (Pythia), and one direction estimate on the full pool instead of one per split (Pythia) or a calibration slice disjoint from the evaluated test items (Qwen). Under both protocols, $r = 2$ moves residual reproduction little on Pythia-2.8B (0.0965 to 0.1118 in Table 10) and substantially on Qwen2.5-7B (0.515 to 0.770 in Table 11).

Scope of the consecutive-gate recommendation. On Pythia-2.8B, $r = 2$ retains approximately $49.0/52.3 = 93.7\%$ of the suppression achieved by $r = 1$, using the rounded entries in Table 13. On the Qwen2.5-7B cohort it retains only about one third. The Pythia ratio should therefore not be generalized to other settings. We treat r as an operating-point choice: $r = 2$ is suitable only when reducing incidental intervention is a priority and its residual reproduction rate is acceptable on validation data.

D.4 Token classes, mask controls, and causal tests

The mask ablation uses 505 Pythia-2.8B sequences and rank-one C with $\eta = 0.1$ and $\beta = 10$. Each mask restricts the gate to a subset of positions. The oracle mask is a diagnostic only, since it requires knowing which positions are verbatim.

Table 15: Mask ablations on Pythia-2.8B (rank-one C, $\beta = 10$, $\eta = 0.1$). "No X" masks exclude positions of class X from the gate.

Mask	Exact	PPL	Number	Entity
Full gate	0.085	24.69	0.324	0.323
No whitespace/punctuation	0.093	24.31	0.377	0.349
No function words	0.089	24.26	0.340	0.361
No numbers	0.089	24.58	0.340	0.321
No entities	0.091	24.48	0.330	0.357
No numbers/entities/function words	0.099	23.95	0.358	0.392
Multi-position runs only	0.091	24.02	0.340	0.345
Runs of length ≥ 3 only	0.101	23.67	0.369	0.358
Oracle verbatim positions	0.097	22.38	0.378	0.390

For the counterfactual single-token analysis, we sample 200 sequences and classify the outcomes for the 111 sequences that the unmodified model reproduces. An outcome is labeled harmful if it loses a

reference number or entity and has low semantic similarity; this is a heuristic rather than a blinded human judgment. The fractions in Table 16 are conditional on the 111 reproduced sequences. They are distinct from the correct-to-incorrect rates on legitimate cohorts, and should not be compared without aligning the population and denominator.

Table 16: Counterfactual single-token classifications (rank-one C, $\beta = 10$, $\eta = 0.1$) with 95% bootstrap intervals.

Class	Fraction	95% CI
Redundant	0.757	[0.682, 0.836]
Generic/interchangeable	0.126	[0.068, 0.180]
Harmful	0.081	[0.038, 0.127]
Verbatim-critical	0.036	[0.000, 0.062]

Matched-budget control. At matched trigger rates, the $r = 2$ gate reduces reproduction by 49.0 percentage points versus 45.4 for a higher-threshold $r = 1$ gate, with correct-token alteration of 3.34% versus 3.33%.

D.5 Threshold calibration under distribution shift

Meaning of the nominal exceedance rate. Let $z = |u^\top h|$ for W and $z = u^\top h$ for the one-sided C/C-M gate, and let $\widehat{F}_{\mathrm{cal}}$ be its empirical calibration distribution. We set

$$\epsilon(\eta) = \widehat{F}_{\mathrm{cal}}^{-1}(1 - \eta).$$

The empirical exceedance rate is approximately η, up to finite-sample quantile conventions and ties. This is not a measured false-positive rate: calibration positions do not supply token-level memorization ground truth, and the distribution of responses may change during generation.

Effect of varying η. For fixed directions and a fixed sequence of unmodified activations, $\eta_2 > \eta_1$ implies $\epsilon(\eta_2) \leq \epsilon(\eta_1)$, so the eligible set of positions cannot shrink. The same set-inclusion property holds for a fixed consecutive-run requirement on that fixed response sequence. For W, lowering the threshold admits additional correction terms; for C/C-M, it also increases the excess coefficient $[u^\top h - \epsilon]_+$ at already-active positions. Neither observation guarantees monotone suppression or utility in autoregressive generation, because interventions change future states and outputs. The run-level trigger probability need not equal η^r, since consecutive responses are dependent.

The real-document sweep in Table 21 illustrates the trade-off. For healthcare at $\beta = 1$ and $r = 1$, increasing η from 0.05 to 0.1 reduces reproduction from 0.233 to 0.053 while control PPL rises from 2.019 to 4.356. With $r = 2$, reproduction decreases from 0.840 to 0.653 and PPL rises from 1.138 to 1.344. These are setting-specific operating points, not a monotonicity guarantee. Figure 9 varies the threshold directly; its horizontal axis should not be interpreted as the quantile parameter η.

Transfer of a calibrated threshold. Table 17 reports teacher-forced exceedance rates of the C probe with a threshold set at the 90th percentile of signed responses on Pile text. The threshold attains its nominal rate on held-out Pile text, but the rate shifts with the evaluation distribution, from 0.038 on number-heavy answers to 0.127 on UltraChat instructions.

Table 17: Probe exceedance rates under distribution shift with a Pile-calibrated threshold (nominal rate 0.1).

Evaluation text	Positions	Exceedance
Pile (held-out)	31600	0.1006
UltraChat instructions	73388	0.1265
Long context (600 tokens)	86580	0.1132
Entity-heavy text	61668	0.1068
GSM8K	31096	0.0572
Number-heavy answers	29381	0.0380

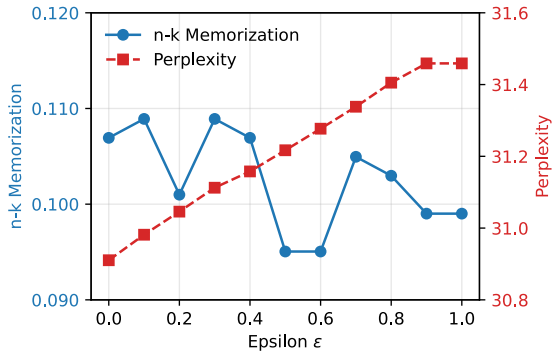


Figure 9: Threshold ablation on Pythia-2.8B ($r = 1$).

D.6 Gated versus ungated results

Table 18 lists the values behind Figure 5. The ungated variant uses the same calibrated directions but removes the indicator and applies the correction at every position. Both variants reach 0.0% memorization on the TinyMem language model at every noise level, and gating lowers PPL by 2.4–4.1 points.

Table 18: Gated versus ungated intervention on the TinyMem language model under increasing numbers of noisy training samples. Memorization is 0.0% for both variants.

Noisy samples	Gated PPL ↓	Ungated PPL ↓	Δ
500	58.97	62.17	−3.2
800	60.40	62.90	−2.5
1000	59.37	63.17	−3.8
1300	62.20	65.10	−2.9
1600	61.60	65.70	−4.1
1900	63.30	66.30	−3.0
2200	66.40	68.80	−2.4
2500	66.90	70.50	−3.6

E Additional models, domains, and utility measures

E.1 Qwen2.5-7B accuracy and judge evaluation

Table 19 reports per-seed reproduction and task accuracy on Qwen2.5-7B with C-M, $r = 1$, and $\eta = 0.1$. Decoding is greedy, so seeds index training and calibration runs rather than sampling.

Two judges score 150 responses per condition on a 1–10 scale: a Qwen judge, which shares the target's model family, and a Llama judge, which does not. We test equivalence with TOST at a ± 0.5 margin; judge equivalence does not imply equal task accuracy. At $\beta = 3$, the Qwen judge supports equivalence for GSM8K seed 1 and UltraChat but not for GSM8K seed 2, and stronger interventions incur larger quality costs. Table 20 lists the Qwen-judge conditions.

For GSM8K seed 1, the Llama judge reports mean change $+0.12$ at $\beta = 3$ and -0.413 at $\beta = 10$, with TOST p values 7.1×10^{-7} and 0.244. Judge-rated quality and final-answer accuracy can therefore disagree, and neither indicates zero utility cost.

E.2 Real-document domains

Qwen2.5-7B is LoRA [28] fine-tuned on 150 documents per domain, duplicated eight times for six epochs. The corpora are PubMedQA contexts, financial-news articles, and ML-arXiv abstracts. Reproduction checks the first 50 continuation tokens after a 50-token prefix under greedy decoding.

Table 19: Qwen2.5-7B on GSM8K: reproduction and accuracy per seed ($r = 1$). Rows with – are the unmodified model.

Seed	β	Reproduction	Accuracy
1	–	0.884	0.593
1	1.0	0.833	0.597
1	3.0	0.513	0.543
1	10.0	0.246	0.460
2	–	0.854	0.597
2	1.0	0.793	0.543
2	3.0	0.416	0.467
2	10.0	0.138	0.350
44	–	0.839	0.587
44	1.0	0.829	0.573
44	10.0	0.336	0.473

Table 20: Qwen2.5-7B-Instruct judge scores (C-M, $r = 1$). Change is relative to the unmodified model; intervals and TOST p-values come from paired tests.

Setting	β	Score	Change	95% CI	p_{TOST}
GSM8K (seed 1)	1.0	7.94	-0.23	[-0.52, 0.05]	0.0361
GSM8K (seed 1)	3.0	8.02	-0.15	[-0.49, 0.17]	0.0203
GSM8K (seed 1)	10.0	7.07	-1.11	[-1.56, -0.65]	0.9958
GSM8K (seed 2)	1.0	7.71	-0.25	[-0.58, 0.09]	0.0758
GSM8K (seed 2)	3.0	7.23	-0.73	[-1.13, -0.35]	0.8703
GSM8K (seed 2)	10.0	6.47	-1.49	[-1.91, -1.05]	1.0000
UltraChat	3.0	6.55	-0.23	[-0.47, 0.00]	0.0140
UltraChat	10.0	5.87	-0.91	[-1.19, -0.63]	0.9973

Table 21 reports the full grid for C-M (rank one, layer 21, ϵ the $(1-\eta)$ quantile of probe responses on the control documents) with $r = 1$ and $r = 2$ over η and β; the two gates share the adapter, directions, threshold, and β of each cell, and no single operating point is a universal default. Because the clean control documents are also used to build the directions, control PPL is not a fully independent test measure.

For example, at $(\eta, \beta) = (0.05, 1)$, finance reproduction drops from 0.933 to 0.221 with PPL 2.302, and at $(0.05, 10)$, academia reproduction drops from 0.919 to 0.302 with PPL 2.011 ($r = 1$). Requiring two consecutive exceedances ($r = 2$) lowers the trigger rate by a factor of 3.3–12.8 and keeps control PPL between 1.14 and 1.81, but it retains only part of the suppression. At $(\eta, \beta) = (0.1, 10)$, reproduction is 0.600, 0.564, and 0.477 with $r = 2$ (healthcare, finance, academia), compared with 0.027, 0.054, and 0.094 with $r = 1$.

E.3 Synthetic PII: duplication and extraction templates

Synthetic account records contain names with associated contact and account fields; this evaluation is separate from the real-document domains. One hundred fictional records (name, email on a reserved example domain, fictional phone number, account ID) are mixed, with 1, 3, or 8 copies each, into the 1,000-item GSM8K SFT set of Qwen2.5-7B (LoRA, six epochs). Extraction prompts each record with its training request and decodes 80 tokens greedily; a record counts as fully extracted when its email, phone number, and account ID all appear, and email extraction checks the email alone. Accuracy is GSM8K final-answer accuracy on the first 300 test items. Directions are rank-one C-M at layer 21, accumulated over answer tokens, and ϵ is the 90th percentile ($\eta = 0.1$) of probe responses on the answer tokens of the first 400 GSM8K test items. Thus, the reported utility set overlaps with threshold-calibration data; this is not a fully independent evaluation of the calibrated operating point. Table 22 reports $r = 1$ and $r = 2$ with the same adapter, directions, ϵ, and β, including weak suppression under heavy duplication and utility loss under strong steering. These canary results do not constitute a privacy guarantee.

With $r = 1$ and 8 copies at $\beta = 10$, full-record extraction remains 0.84 and email extraction 1.00; with 3 copies, strong suppression coincides with accuracy falling from 0.61 to 0.22. Requiring

Table 21: Real-document domains, C-M with $r = 1$ and $r = 2$ (same directions, ϵ, and β per row). Exact-50 is exact match of the first 50 continuation tokens, PPL is measured on control documents, and trigger is the fraction of positions at which the gate corrects. Rows with – are the unmodified model.

			Exact-50		Control PPL		Trigger	
Domain	η	β	$r = 1$	$r = 2$	$r = 1$	$r = 2$	$r = 1$	$r = 2$
healthcare	–	–	0.913		1.089		–	
healthcare	0.05	1	0.233	0.840	2.019	1.138	0.049	0.004
healthcare	0.05	3	0.153	0.820	3.213	1.182	0.049	0.004
healthcare	0.05	10	0.133	0.813	4.332	1.215	0.050	0.004
healthcare	0.1	1	0.053	0.653	4.356	1.344	0.097	0.015
healthcare	0.1	3	0.040	0.613	12.173	1.611	0.097	0.015
healthcare	0.1	10	0.027	0.600	20.885	1.768	0.098	0.015
finance	–	–	0.933		1.098		–	
finance	0.05	1	0.221	0.785	2.302	1.188	0.049	0.005
finance	0.05	3	0.168	0.752	3.655	1.228	0.049	0.005
finance	0.05	10	0.128	0.738	4.942	1.260	0.052	0.005
finance	0.1	1	0.074	0.617	5.303	1.425	0.095	0.016
finance	0.1	3	0.060	0.591	13.017	1.626	0.101	0.017
finance	0.1	10	0.054	0.564	22.502	1.776	0.114	0.019
academia	–	–	0.919		1.464		–	
academia	0.05	1	0.919	0.933	1.473	1.468	0.049	0.011
academia	0.05	3	0.591	0.846	1.594	1.496	0.050	0.011
academia	0.05	10	0.302	0.685	2.011	1.567	0.050	0.011
academia	0.1	1	0.886	0.906	1.485	1.472	0.097	0.029
academia	0.1	3	0.362	0.685	1.833	1.561	0.100	0.029
academia	0.1	10	0.094	0.477	3.006	1.814	0.100	0.029

Table 22: Synthetic PII extraction by number of copies (C-M, $\eta = 0.1$) with $r = 1$ and $r = 2$. Full-record and email columns are extraction rates over 100 records; accuracy is GSM8K test accuracy; trigger is the fraction of real (non-padding) token positions at which the gate corrects. Rows with – are the unmodified model.

Copies	β	r	Full record	Email	Accuracy	Trigger
1	–	–	0.00	0.02	0.597	–
1	3.0	1	0.00	0.03	0.337	0.069
1	3.0	2	0.00	0.02	0.470	0.016
1	10.0	1	0.00	0.02	0.340	0.062
1	10.0	2	0.00	0.02	0.510	0.014
3	–	–	0.77	0.86	0.610	–
3	3.0	1	0.59	0.73	0.550	0.096
3	3.0	2	0.76	0.83	0.567	0.024
3	10.0	1	0.01	0.04	0.220	0.104
3	10.0	2	0.35	0.38	0.467	0.031
8	–	–	1.00	1.00	0.600	–
8	3.0	1	1.00	1.00	0.587	0.067
8	3.0	2	1.00	1.00	0.583	0.024
8	10.0	1	0.84	1.00	0.473	0.065
8	10.0	2	1.00	1.00	0.570	0.023

two consecutive exceedances ($r = 2$) lowers the trigger rate by a factor of 2.8–4.6 and recovers accuracy (0.467 instead of 0.220 with 3 copies at $\beta = 10$), but it suppresses much less: with 3 copies, full-record and email extraction are 0.35 and 0.38 at $\beta = 10$ (0.01 and 0.04 with $r = 1$) and 0.76 and 0.83 at $\beta = 3$ (baseline 0.77 and 0.86), and with 8 copies full-record extraction stays at 1.00 at both strengths.

We also vary the extraction prompt. Five templates request the record in training format, a paraphrased lookup, all stored customer details, the email alone, or the phone/account ID (100 records each, 3 copies). Across templates that elicit the full record, baseline full-record extraction is 0.77–0.79 and email extraction 0.86–0.87. With $r = 1$, $\beta = 3$ reduces these to 0.47–0.59 and 0.72–0.73, and $\beta = 10$ to at most 0.01 and 0.04. With $r = 2$, they are 0.74–0.78 and 0.80–0.84 at $\beta = 3$, and 0.33–0.43 and 0.35–0.46 at $\beta = 10$. The email-only template does not ask for the full record, so its full-record rate is zero even though the email itself is extracted (0.87 at baseline; 0.76 and 0.05 with $r = 1$ and 0.82 and 0.29 with $r = 2$ at $\beta = 3$ and 10).

E.4 Approximate and semantic reproduction

Zeng et al. [71] study memorization during fine-tuning and distinguish verbatim, paraphrase, and idea memorization. Idea memorization can reveal information from a training continuation absent from the prompt without reproducing its wording, for example by condensing that information. Our evaluation below is an exploratory semantic-overlap analysis; it does not reproduce their detection pipeline or establish mitigation under their metric.

The Pythia semantic sweep uses rank-one sign-canonicalized contrastive directions, layer 22, nominal exceedance 0.1, and a single-position gate. Semantic similarity is a MiniLM embedding-cosine proxy. We use "semantic-only proxy" for the fraction of outputs with cosine greater than 0.8 without exact reproduction. This proxy does not by itself identify exposure-caused information leakage or distinguish it from legitimate semantic similarity.

Table 23: Exact and semantic reproduction on Pythia-2.8B across steering strengths. Weak steering increases the semantic-only proxy, so exact suppression does not imply semantic suppression.

β	Exact	4-gram precision	Cosine	Semantic-only proxy
Baseline	0.535	0.811	0.913	0.307
1.0	0.248	0.545	0.785	0.356
3.0	0.141	0.425	0.690	0.299
5.0	0.107	0.365	0.641	0.259
15.0	0.081	0.307	0.592	0.218
20.0	0.077	0.286	0.575	0.198
30.0	0.046	0.256	0.545	0.198

At strength 1, exact reproduction decreases but the semantic-only proxy rises from 0.307 to 0.356. Stronger correction reduces all overlap proxies, at a potential cost in generation quality. A lower exact-match rate can therefore coexist with more non-exact, semantically similar outputs. These results do not support a general claim that GSS prevents idea memorization. Evaluating that claim requires a dedicated exposure-sensitive protocol and an independent utility assessment.

Tables 24 and 25 use a separate protocol with a similarity threshold of 0.5 for approximate reproduction.

Table 24: TinyMem exact and approximate reproduction ($r = 1$).

Metric	Before	After	Delta
Exact memorized %	23.97	0.00	-23.97
Approx-only memorized %	76.03	3.64	-72.39
ROUGE-L mean	0.9664	0.3749	-0.5916

Table 25: Pythia-2.8B exact and approximate reproduction ($r = 1$).

Metric	Before	After	Delta
Exact memorized %	46.14	8.91	-37.23
Approx-only memorized %	42.18	34.06	-8.12
ROUGE-L mean	0.8529	0.4740	-0.3790

E.5 Extraction attacks

Table 26 lists the attack settings in which steering reduced attack success. It is a subset of the evaluated grid and does not indicate uniform robustness across attacks.

Table 26: Attack settings in which steering reduces attack success rate (a subset of the evaluated grid).

Attack template	Prefix fraction	Base ASR (%)	Steered ASR (%)	Reduction (%)
Instruction-0	0.2	9.7	5.1	47
Instruction-0	0.5	18.5	9.2	50
Instruction-2	0.3	16.4	5.1	69
Paraphrase-0	0.3	16.4	5.1	69
Paraphrase-0	0.5	18.5	7.2	61

Table 27: Best-of-8 extraction on 145 sequences ($r = 1$). Exact-match ASR is already zero at baseline, so overlap-based ASR is the informative outcome.

Template	Prefix Fraction	# Examples	Metric	Base	Steered
Raw prefix	0.5	145	Exact Match ASR	0.00	0.00
Raw prefix	0.5	145	ASR@50% Token Overlap	54.48	0.00
Raw prefix	0.5	145	ASR@80% Token Overlap	2.07	0.00
Raw prefix	0.5	145	Mean Token Overlap	0.4469	0.1390
Raw prefix	0.5	145	Mean LCP Ratio	0.2855	0.0172
Raw prefix	0.5	145	Mean ROUGE-L	0.6404	0.3792

Interpreting zero successes. Under an independent sequence-sampling model, zero successes among the 145 sequences in Table 27 gives a two-sided 95% exact binomial upper limit of $1 - 0.025^{1/145} \approx 2.51\%$ for success within the specified attack budget. The eight attempts on each sequence are not 1,160 independent sequence trials. This conditional interval is not a bound for arbitrary prompts, adaptive attacks, or another sequence population.

F Further ablations and additional results

F.1 Probe discrimination and stability

Table 28 reports how well the probe subspace separates memorized from non-memorized sequences, and Table 29 its stability under resampling. Stability alone does not establish a causal memorization subspace or transfer across domains.

F.2 Long-form generation

We compare 400-token greedy outputs of Llama-3.2-1B before and after intervention. ROUGE-L between the two outputs measures their similarity, not correctness, and Distinct-n measures lexical diversity.

F.3 Layer selection

Layer-dependent probe statistics can guide calibration choices, but the best layer for a signal statistic need not optimize the suppression–utility trade-off. Figure 10 shows the per-layer probe statistics.

Table 28: Sequence-level discrimination performance of the whitened probe subspace on memorized vs. non-memorized sequences.

Model	Metric	Rank 4	Rank 8
Qwen3-0.6B	AUC	0.783	0.811
Qwen3-0.6B	Spearman	0.530	0.603
Llama-3.2-1B	AUC	0.899	0.929
Llama-3.2-1B	Spearman	0.771	0.839

Table 29: Stability of the learned probe and steering directions across resampled training subset fractions.

Direction	cos
u (probe)	0.999 ± 0.001
v (steer)	0.943 ± 0.067

F.4 Bootstrap stability of activation subspaces

For two orthonormal bases $U_i, U_j \in \mathbb{R}^{d\times K}$, define $S_{ij} = \|U_i^\top U_j\|_*/K \in [0,1]$. We compare bootstrap PCA subspaces estimated on positive-signal positions, nonpositive-signal controls, and random Gaussian subspaces. These PCA subspaces differ from the gradient-weighted GSS directions. Higher agreement across resamples indicates stability in this setting; it does not show that all memorization is low-dimensional.

F.5 Utility-constrained modern-model results

Table 31 reports, for each method, the largest reproduction reduction among operating points whose normalized utility is at least 0.9. GSS gives the largest reduction on GSM8K for both models, while several unlearning baselines achieve larger reductions on UltraChat.

F.6 Qualitative examples

Figure 12 shows contract-QA outputs before and after steering. Repetition, malformed strings, or unusual numbers do not by themselves indicate memorization, which requires a match to a training sequence.

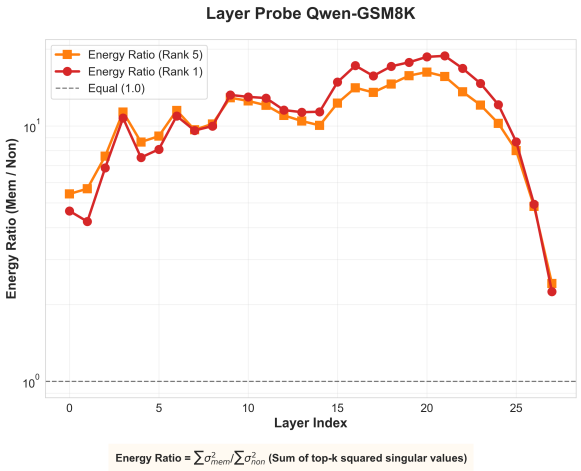


Figure 10: Per-layer probe statistics on Qwen3-0.6B, GSM8K.

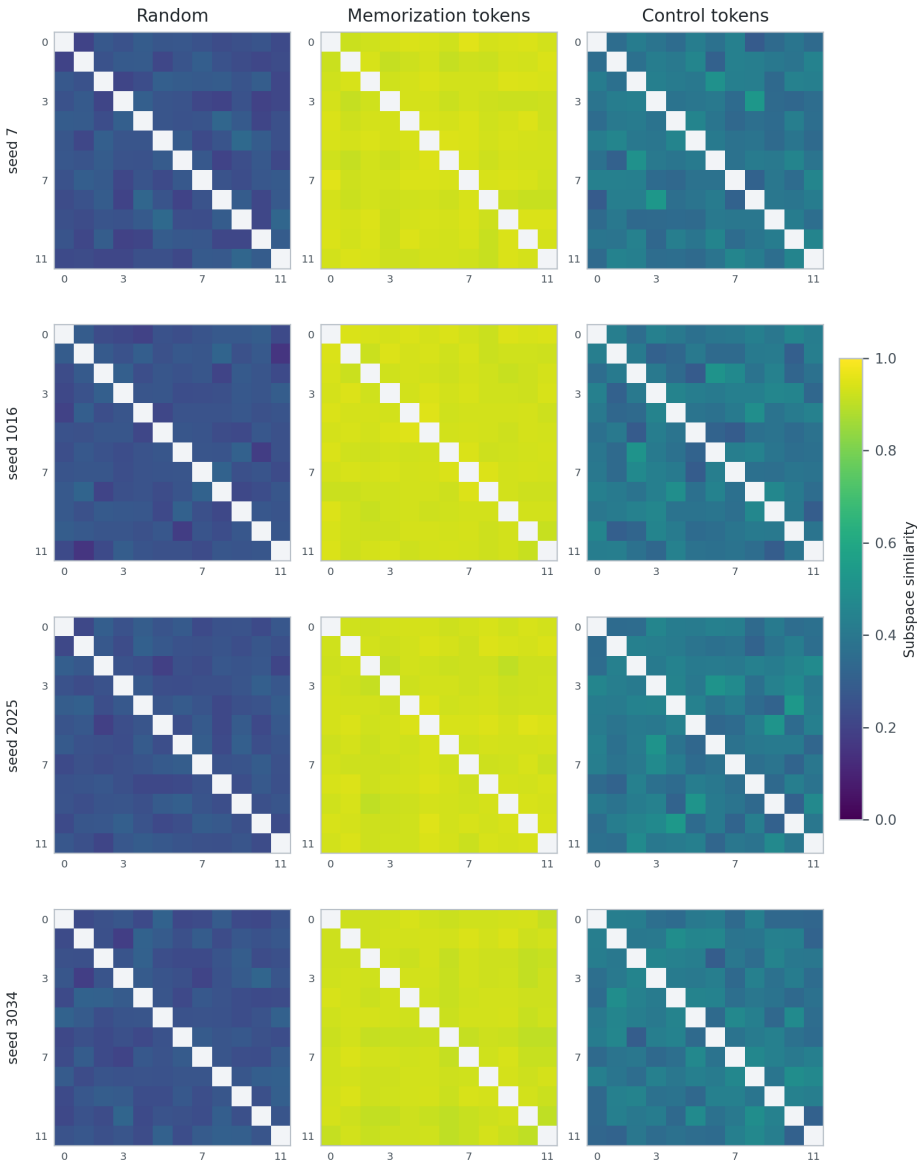


Figure 11: Bootstrap similarity of PCA activation subspaces.

Prompt:

moved.

You can now find them here.
This forum is now closed to new posts, but you can browse existing content.
You can find

Before Steering

out more information about the move and how to open a new account (if necessary) here. If you get stuck you can get support by emailing forums@sitepoint.com If this is your first visit, be

After Steering

out more information about the move and how to open a new account here. If you have already joined, please log in. If this is your first visit, be sure to check out the FAQ by clicking the link

Figure 12: Contract-QA outputs before and after steering.

Table 30: Long-form generation on Llama-3.2-1B: ROUGE-L between baseline and GSS outputs, and change in Distinct-n.

Condition	ROUGE-L (Base, GSS)	Distinct-2 Δ	Distinct-3 Δ
Clean (unseen test)	0.835	+0.009	+0.004
Memorized (train prefix)	0.685	+0.023	+0.031

Table 31: Largest reproduction reduction (%) subject to normalized utility of at least 0.9.

Category	**Method**	**Llama (Red. %)**		**Qwen (Red. %)**	
		UltraChat	**GSM8K**	**UltraChat**	**GSM8K**
Unlearning	GA	95.5	4.5	3.3	0.0
	Task Vector	77.0	25.1	66.0	25.4
	DPO	95.7	1.3	82.4	7.3
	SimNPO	95.5	2.2	78.8	7.2
	NPO	88.8	2.2	71.5	6.9
Editing	ROME	38.0	7.8	8.6	14.5
	MEMIT	7.5	2.2	2.2	8.7
	Balanced Subnet	34.8	1.4	1.0	12.8
	LUNAR	30.0	9.0	42.7	1.0
	Ours	48.0	33.5	51.4	35.2

G Methodological connections

G.1 Low-rank updates

Ignoring gating, a linear activation update $h' = (I - \beta VU^\top)h$ placed immediately before a linear map W gives $Wh' = (W - \beta WVU^\top)h$. This is algebraically a rank-at-most-K weight modification at that location. It does not imply equivalence to a trained LoRA adapter throughout a nonlinear Transformer. A gate makes the update input-dependent, and a ReLU excess coefficient also includes a threshold-dependent term. Reversibility means the intervention can be disabled; it does not certify removal of stored information from the unchanged weights.

G.2 Relation to discriminant analysis

Classical discriminant analysis uses class-scatter statistics. Here the coupling matrix combines activations and log-likelihood gradients (negated CE gradients; Appendix A.3), while the whitened configuration normalizes an uncentered low-signal second moment. This asymmetric coupling motivates separate probe and steer directions, and setting $u = v$ does not in general recover mean-difference steering.

G.3 Unlearning and editing

Machine unlearning studies removal of the influence of designated training examples; surveys and methods examine this objective across learning settings [45, 70, 13, 58, 69, 30, 61]. Knowledge editing instead studies targeted changes to model knowledge or behavior [27, 35, 34]. Unlearning and editing provide objective-related comparisons for reducing specified reproduction behavior. Activation steering provides mechanism-related comparisons. GSS can reduce reproduction while active without changing the underlying weights; this is not exact unlearning or a retraining-equivalence guarantee.

H Broader impact and limitations

H.1 Societal implications

Reducing verbatim reproduction may lower exposure to training-data extraction and unauthorized reproduction of sensitive or copyrighted content under the evaluated protocols. GSS does not remove information from the model weights, and these experiments do not establish comprehensive privacy protection, fairness improvements, or out-of-distribution robustness.

We acknowledge potential dual-use concerns: techniques mitigating memorization could obscure training data sources, complicating model auditing; selective memorization reduction could be misused to hide specific training data while maintaining overall performance; and over-aggressive mitigation might remove useful factual knowledge.

H.2 Limitations

Beyond the proxy-related limitations discussed in Appendix A.2, our approach has several practical limitations:

1. **Reference-model dependence**: Computing ω_t requires a domain-competent, lower-exposure reference. The original pretrained checkpoint is not required, but a suitable alternative may be unavailable; arbitrary model disagreement is not sufficient.
2. **Semantic leakage**: Exact-match suppression does not guarantee mitigation of paraphrase or idea memorization (Appendix E.4).
3. **Calibration and utility**: Domain control PPL and the PII utility set are not fully independent of calibration. Results should not be interpreted as independently validated deployment guarantees.
4. **Calibration cost**: Direction estimation requires activations, gradients and a matrix decomposition. Online application cost excludes these one-time computations and must be reported separately.
5. **Layer selection**: Effectiveness depends on choosing appropriate layers, which may require empirical validation.
6. **Run requirement**: Requiring two consecutive exceedances ($r = 2$) lowers the trigger rate in every evaluated setting, and the share of suppression it keeps depends on the setting. On Pythia-2.8B, $r = 2$ keeps 49.0 of the 52.3 points of verbatim reduction of $r = 1$ (Table 13). On fine-tuned Qwen2.5-7B, it keeps 14.8 of 45.8 points on GSM8K (Table 14); at $(\eta, \beta) = (0.1, 10)$, reproduction on the three real-document domains is 0.600, 0.564, and 0.477 with $r = 2$ against 0.027, 0.054, and 0.094 with $r = 1$ (Table 21); and with eight PII copies at $\beta = 10$, full-record extraction is 1.00 with $r = 2$ against 0.84 with $r = 1$ (Table 22). In these fine-tuning settings, r is a parameter that trades suppression for utility and is chosen per setting.

H.3 Future directions

Our results suggest several promising directions for future work. First, the current study focuses on text-only language models, while memorization is also a central concern in vision-language and other multi-modal systems [65]. Extending GSS to multi-modal models would require identifying whether memorization-relevant activations are shared across modalities or localized within modality-specific encoders, such as visual tokenizers, audio encoders, or cross-modal projection layers. This would also allow us to study whether the concentration of positive likelihood signal observed in language modeling persists when the model conditions on images, speech, or structured visual inputs.

Additionally, memorization behavior may differ substantially across languages and writing systems [52]. Multilingual models are trained on highly imbalanced corpora, and low-resource languages may exhibit different memorization patterns due to duplicated data, translation artifacts, or uneven tokenization quality. Extending our analysis to multilingual settings would test whether a single shared memorization subspace is sufficient, or whether language-specific probes and steering directions are needed. Together, these directions would broaden GSS from a text-only mitigation method into a more general framework for context-aware memorization control.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state our three contributions: (1) the empirical characterization of memorization as sparse and heavy-tailed at the token level (Section 3, Fig. 1), (2) the GSS method with its probe–steer decomposition (Section 4), and (3) the closed-form derivation of the optimal probe–steer pair (Theorem 4.1). All claims are supported by experiments in Section 5 across four benchmarks (TinyMem, Pythia-2.8B/6.9B, Llama-3.2-1B and Qwen3-0.6B fine-tuned on GSM8K and UltraChat), summarized in Tables 1 and 2.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations are discussed in Appendix H, including reference-model dependence, computational overhead of the SVD, inference overhead from gating, threshold sensitivity, and layer selection. Additional proxy-related limitations (population-level signal, behavior under adversarial fine-tuning, requirement of a reasonable reference distribution) are discussed in Appendix A.2.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.

- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: All theoretical results are stated with assumptions and proven. Theorem 4.1 (Closed-Form Surrogate Optimum) is proven in Appendix A.4. A fixed-direction gate perturbation bound is derived in Appendix A.5. The derivation of the token-level signal ω_t is provided in Appendix A.1, and the steering strength β used in Eq. (8) is derived in Appendix A.7.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Section 5 specifies the benchmarks, baselines, and metrics. Section 4 fully specifies the GSS calibration and intervention procedure, and Appendices B.2, B.1, and B.2 detail the baseline configurations, memorized data extraction procedure, and hyperparameter search spaces.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example

(a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
(b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: Our code is not yet publicly available; we plan to release it after cleanup. We use publicly available models (Pythia-2.8B/6.9B, Llama-3.2-1B, Qwen3-0.6B) and publicly available datasets (TinyMem, GSM8K, UltraChat, The Pile). The full method is specified in Section 4. All steps for memorized data extraction are documented in Appendix B.1.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Experimental settings are described in Section 5 (benchmarks, baselines, metrics). Detailed hyperparameter configurations and grid-search spaces are provided in Appendix B.2, baseline-specific optimization settings (optimizer, learning rates, batch sizes, epochs) are documented in Appendix B.2, and the GSS-specific hyperparameters (rank K, threshold ϵ, variance budget δ) are described in Section 4 and ablated in Section 5 as well as Appendices F.3.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Results on TinyMem (Table 1) are averaged over 3 random seeds, as stated in the table caption. The reference-model ablation (Section 5.8, Figure 6) reports memorization rates across multiple reference checkpoints, and the stability analysis in Appendix F.1 reports cosine-similarity statistics (mean $\pm$ std) across resampled subsets. Bootstrap stability of the memorization-aligned subspace is reported in Appendix F.4.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Wall-clock runtimes are reported in Tables 1 and 2 (inference-stage time).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: The research uses only publicly available models and datasets and does not involve human subjects or personally identifiable data collection. Anonymity is preserved throughout the submission. We discuss broader impacts and dual-use concerns in Appendix H.1.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: Appendix H.1 discusses both positive impacts (privacy via reduced verbatim reproduction of training data, copyright protection, fairness via reduced biases from memorized patterns, robustness via better generalization) and negative impacts (potential to obscure training data sources and complicate auditing, possible misuse to selectively hide specific training data, risk of over-aggressive mitigation removing useful factual knowledge).

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [Yes]

Justification: Our work releases an inference-time mitigation method, not a new pre-trained model or scraped dataset. The method itself reduces rather than increases misuse risk by suppressing verbatim reproduction of training data. All experiments use publicly released models and datasets.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All models (Pythia, Llama-3.2, Qwen3) and datasets (TinyMem [57], GSM8K [67], UltraChat [17], The Pile [22]) used in the paper are properly cited. All assets are used under their original public licenses and consistent with their intended research use.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not release new datasets or pretrained models. We plan to release code implementing GSS (Section 4) with documentation after it has been cleaned up; no new pretrained-model or dataset asset is introduced.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or human subjects. All experiments are conducted on publicly available models and datasets.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve human subjects or any study participants, so no IRB approval is required.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are the *subject of study* (we evaluate memorization in Pythia, Llama-3.2, and Qwen3), but they are not used as a methodological component of the proposed GSS algorithm. Any LLM use for writing or editing assistance does not affect the core methodology, scientific rigor, or originality of the research.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.