PROBE-GEOMETRY ALIGNMENT: AUDITING AND SUPPRESSING REPRESENTATIONAL MEMORIZATION IN LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Unlearning methods for language models are judged by what the model says. Prompt the model with the opening words of a target passage, and if it no longer completes the passage, the content is reported as removed. A model can pass that test and still encode the passage. We measure what is left with a probe that has to generalize. A linear classifier is fitted on the hidden states of several memorized sequences, then tested on a memorized sequence it has never seen. To succeed, it must read a property of memorization itself rather than a particular string. It succeeds on every model we tested. Across three pretrained architectures, the memorization-specific gap runs from $+0.19$ to $+0.35$. It is stable across the Pythia family from 70M to 1B parameters, and it holds on a pool whose membership is verified by corpus lookup rather than by likelihood. The direction the probe reads is causally separable from the machinery that recites the text. No behavioral erasure we applied removes it, from five published unlearning objectives to weight editing to imitating a model that never memorized, however far the output moves. That separation is what makes an efficient method possible. Probe-geometry alignment (PGA) constrains one scalar per depth, the projection onto the direction the probe reads, and leaves the rest of the residual stream alone. On Pythia-70M it buys 6.6 times as much recall suppression per nat of capability as the best of four isotropic settings on the same recipe. It also suppresses more recall than the behavioral baseline that retains the most capability, while retaining more capability than that baseline, and it carries over to Mistral-7B. Constraining a readout can also reverse it rather than empty it. Raw probe accuracy then falls below chance while the classes stay separable, so an audit that quotes that number records erasure where none occurred. A multi-position variant drives a fresh mean-pooled linear probe to chance at mid depths, at a further cost in capability. The audit and the method share one direction, so a deployment able to fit a probe can state what its deletion claim is worth, and act on the answer.

1 INTRODUCTION

Machine unlearning for language models is evaluated by what the model says. A method is applied, the model is prompted with the opening words of the target text, and if it no longer completes that text the content is reported as removed. Major frameworks, including TOFU (Maini et al., 2024), MUSE (Shi et al., 2025) and WMDP (Li et al., 2024), score unlearning by the model's outputs. Membership-inference work on pretraining data (Shi et al., 2024) asks a related but different question, whether a record was trained on rather than whether it is still encoded.

That criterion is not robust. Benign downstream fine-tuning brings back content that unlearning appeared to remove (Hu et al., 2025), a failure that recurs across eight evaluation axes (Lynch et al., 2024), quantization alone reverses the effect (Zhang et al., 2025b), and suppression does not generalize to semantically related content (Jia et al., 2025). Each result is evidence that something survived, and none of them says what survived, where it sits, or how much of it is left. Deletion is also something an organization may have to demonstrate rather than simply perform (Zhang et al., 2025a), and accountability frameworks ask that such claims rest on evidence an auditor can examine

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

(Raji et al., 2020). We take the auditor to be a third party, a reader with access to the model's weights and activations who fits a classifier to them to test whether the target content is still encoded.

This paper measures what remains directly. Base Pythia-70M reproduces the MIT license preamble almost verbatim from its opening words. We write a register-matched control, a paraphrased license of the same length and style that the model has not memorized, and train a linear probe on hidden states to separate the two. Any probe can do this for a single pair by reading the tokens. The informative question is whether a probe trained on several such pairs classifies a *held-out* memorized sequence it has never seen. If it does, the hidden states carry a feature of memorization itself rather than of any particular string. We call it the cross-sequence memorization signature, and the leave-one-out (LOO) protocol that measures it is the instrument this paper builds.

The instrument reports a strong and reproducible signal, positive on every model tested and size-stable across the Pythia family at five to eight standard deviations above a shuffled-label null. The direction it reads is causally separable from recall: project it out of the residual stream, the running sum a transformer carries from layer to layer, and the local signature collapses while the model keeps reciting the text. The representation therefore carries something the generation test does not reach, and no behavioral erasure we applied removes it, including weight editing (Meng et al., 2023) and distillation (Hinton et al., 2015) from a model that never memorized.

Separating the two axes is what makes the method possible. A probe reads one direction per depth, so an erasure objective need not move the rest of the residual stream, most of which supports the model's other behavior. Probe-geometry alignment constrains that one scalar. This is where its efficiency comes from, and also where a measurement problem appears, because constraining what a probe reads can reverse that reading rather than remove it.

The paper makes four contributions.

1. **A protocol that makes representational memorization measurable** (Sections 3.1, 4.1 and 4.2). The cross-sequence LOO probe asks whether a memorization signature *generalizes* to held-out sequences. It needs neither the training corpus nor a pre-unlearning reference model (Xu et al., 2025), and it tests a specific passage rather than whether a model was unlearned at all (Chen et al., 2026). The signature is positive on all three architectures we test, and the direction it reads is causally separable from recall: projecting it out collapses the local signature from $+0.643$ to $+0.161$ while recall falls by at most 0.116 nats per token.
2. **Evidence that behavioral erasure does not reach the encoding** (Section 4.3). Five published unlearning methods leave the probe at 1.000 on a controlled testbed, a small transformer trained to recall one injected sentence, and four show the same pattern on Pythia-70M while changing the output by up to 62 nats. Weight editing leaves the probe at 1.000, and distillation from a teacher that never memorized erases the behavior at almost no capability cost yet leaves the probe at 0.922.
3. **Probe-geometry alignment (PGA), which constrains only what the probe reads** (Sections 3.2, 4.4 and 4.5). PGA constrains one scalar per depth. On Pythia-70M it suppresses more recall than refusal tuning (IDK), the baseline that keeps the most capability, while also keeping more capability (3.25 against 1.05 nats, at 88.6% against 79.4% retention). An isotropic penalty on the same recipe, layers and budget is 6.6 times less efficient at its best setting and 19.5 times less at PGA's own penalty weight, so the gain comes from the geometry rather than from penalty strength. PGA also suppresses recall on Mistral-7B, and a multi-position variant drives a fresh mean-pooled linear probe to chance at mid depths.
4. **Raw probe accuracy is not a safe erasure criterion** (Section 4.3 and Appendix Q). Alignment can reverse a readout rather than remove it. A probe fitted afterwards then gets almost every label wrong, so its raw accuracy falls toward 0 rather than toward chance, and an auditor quoting it would record erasure where the classes remain separable. Raw accuracy falls below chance at all four scales we test, and of ten configurations audited both ways only the three trained PGA settings change verdict. Reporting orientation-corrected accuracy, $\max(\mathrm{acc}, 1 - \mathrm{acc})$ (Burns et al., 2023), avoids the error, and the point applies to any erasure claim audited with a probe.

Figure 1 summarizes the evidence in three panels: the signature holds across scale, four existing methods move the output and not the signature, and aligning the probe's own direction suppresses

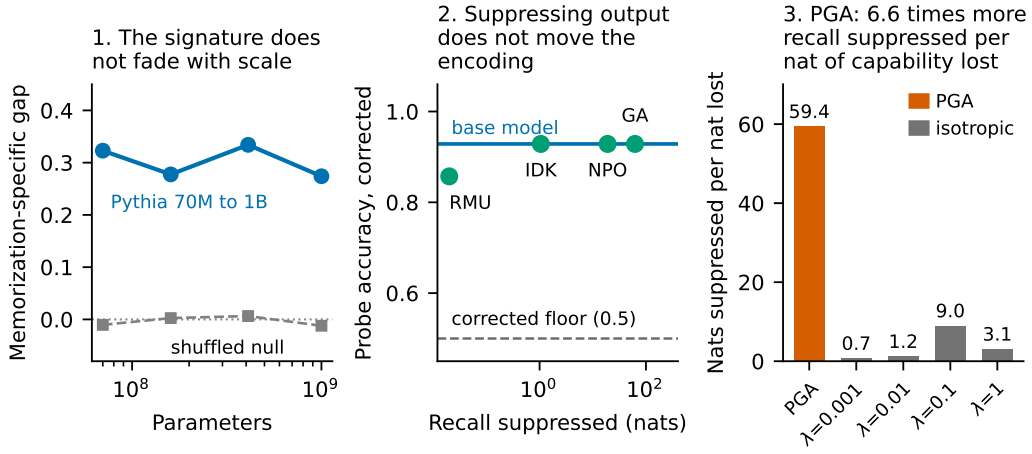

Figure 1: Summary of the three results.

far more recall per nat of capability than acting on the whole residual stream. The four are gradient ascent (GA), negative preference optimization (NPO), representation misdirection for unlearning (RMU) and IDK.

2 Related work

Machine unlearning removes the influence of specific training data without retraining from scratch (Cao & Yang, 2015; Bourtoule et al., 2021). Recipes range from Fisher-information weight scrubbing (Golatkar et al., 2020) and surgical replacement (Eldan & Russinovich, 2023) to NPO (Zhang et al., 2024) and gradient ascent on negative examples (Yao et al., 2024), dozens of which Ren et al. (2025) catalog. What unites them is the evaluation, and by that measure most succeed. The robustness failures cited in Section 1 infer retention indirectly, as do Chen et al. (2025) and Wang et al. (2025), who establish it by exhibiting a recovery procedure. We measure the retained structure directly, at a specified depth against a stated null, which an auditor can run without a recovery procedure in hand.

A linear probe that separates a class lower-bounds what the hidden state carries (Marks & Tegmark, 2024; Hewitt & Manning, 2019), provided its accuracy is read against a null rather than on its own (Belinkov, 2022). A probe that still classifies after unlearning is therefore evidence the structure survived. Knowledge editing localizes factual recall to mid-layer feed-forward modules, where a targeted weight edit changes the output (Meng et al., 2022; 2023), but an intervention that succeeds at one site does not establish that the information was stored only there. Activation steering makes the point from the other direction, shifting behavior with a vector added to the residual stream and no weights touched (Turner et al., 2023). We use causal tracing (Meng et al., 2022) to identify the minimal circuit (Elhage et al., 2021) behind a memorized completion, and the probe to test whether an intervention reached storage or only silenced output.

What this paper measures is a property that transfers between sequences, which is what the question demands when it asks whether *this passage* is still encoded rather than whether *this model* was tampered with. It is also what lets one instrument serve both as an audit and as the target of a suppression objective. Related diagnostics read latent knowledge item by item (Lynch et al., 2024), quantify drift against a pre-unlearning copy (Xu et al., 2025), or train a classifier to detect whether a model has been unlearned at all (Chen et al., 2026). None tests whether the signature generalizes to held-out sequences, and none turns its diagnostic into a training objective. Table 1 sets out the comparison.

Concept erasure supplies the closest methodological neighbors, each removing a concept that a linear classifier could read: the nullspace projection of INLP (Ravfogel et al., 2020), the minimax of R-LACE (Ravfogel et al., 2022), and the closed-form guarantee of LEACE (Belrose et al., 2023).

Table 1: Comparison with prior work.

Study	Measurement	Measurement needs reference model	Cross-sequence transfer	Suppression method
Hu et al. (2025)	Recovery after fine-tuning	No	No	No
Lynch et al. (2024)	Latent-knowledge probe	No	No	No
Zhang et al. (2025b)	Recovery after quantization	No	No	Yes
Jia et al. (2025)	Recall on related content	No	No	No
Chen et al. (2025)	Prompt and activation audits	No	No	No
Chen et al. (2026)	Unlearning-trace classifier	No	No	No
Xu et al. (2025)	Representational drift	Yes	No	No
Belrose et al. (2023)	Fixed-concept recoverability	No	No	Yes
Zhang et al. (2024)	Behavioral recall (NPO)	No	No	Yes
Maini et al. (2024)	Match to retrain (IDK)	Yes	No	Yes
Li et al. (2024)	Hazard question accuracy	No	No	Yes
Wang et al. (2025)	Recall after later fine-tuning	No	No	Yes
This work	Held-out probe transfer	No	Yes	Yes

Those guarantees hold when the concept is fixed and known in advance. Memorization is neither, which is why the method of Section 3.2 refits its direction during training rather than solving once.

3 METHODS

3.1 THE CROSS-SEQUENCE LEAVE-ONE-OUT PROTOCOL

A probe trained to separate one memorized string from one control reaches accuracy 1.000 on every model we tested, and learns the tokens rather than anything about memorization. The LOO design removes that shortcut by requiring the probe to classify a sequence it has never seen.

For each model we build three per-sequence pools on shared identifiers, one key per source text such as `mit_license`. POOL_M holds sequences passing a per-model screen on mean log-probability per token. POOL_C holds a register-matched paraphrase per key, built by substituting distinctive terms so that register and surface form are preserved while the model no longer reproduces the text. Where $\mathrm{POOL}_M[\texttt{mit_license}]$ begins "Permission is hereby granted", $\mathrm{POOL}_C[\texttt{mit_license}]$ begins "Permission is explicitly denied, under all conditions". POOL_D holds a second, further paraphrased variant per key. A shared pool $\mathcal{B}$ of 24 neutral prefixes is prepended at extraction time to supply within-class variance. Thresholds are in Appendix C and the verbatim pools in the supplementary archive.

At each residual-stream depth d, with $d = 0$ the embedding output and one depth per transformer layer thereafter, and for each held-out key, we fit a logistic regression probe on the activations of all remaining keys, labeling POOL_M positive and POOL_C negative, and evaluate on the held-out key's 24 memorized and 24 control variants. Averaging over held-out keys gives the *true LOO* accuracy. Two nulls calibrate it. The *shuffled-label* null permutes training labels within each fold (Hewitt & Liang, 2019). The *pure-distinguishability* null repeats the procedure with POOL_C positive and POOL_D negative, both screened as non-memorized, and measures how separable two register-matched paraphrases are from one another. That second null matters most, because it measures the nuisance separability a naive reading would mistake for memorization. The memorization-specific gap is their difference,

$$g_d \;=\; \mathrm{LOO}_{\mathrm{true}}(d) \;-\; \mathrm{LOO}_{\mathrm{pure}}(d). \tag{1}$$

We summarize a model by the mean of g_d over transformer depths, excluding the embedding output, where token identity is trivially readable. Throughout, the *base model* is the model before any unlearning or alignment. *Recall suppression* is its mean $\log P/\mathrm{tok}$ on the memorized pool minus the same quantity after an intervention, in nats per token, the natural-logarithm unit of log-probability. *Held-out cost* is the corresponding drop in mean $\log P/\mathrm{tok}$ on held-out text, also per token, so the ratio of the two is dimensionless.

A linear readout is defined only up to sign, since negating the weight vector costs nothing and reverses every prediction. Where the question is what an adversary can recover, we report accuracy orientation-

corrected, $\mathrm{acc}^{\pm} = \max(\mathrm{acc}, 1 - \mathrm{acc})$, the convention Burns et al. (2023) use for a probe whose sign is not fixed. Because the correction folds every reading below 0.5 upward, an uninformative readout scores above 0.5 in expectation. Wherever we ask whether a corrected accuracy is above chance, we compare it against a permutation null of 200 label permutations (Ojala & Garriga, 2010) rather than against 0.5, and report the null's 95th percentile. For the statistic used in Sections 4.3 to 4.5, the maximum corrected logistic-regression accuracy over depths two to six on Pythia-70M, that percentile is 0.929. Eq. (1) subtracts raw accuracies, since correcting a chance-centered null would inflate it (Appendix C).

3.2 Probe-geometry alignment

A suppression method has to move the readout of Section 3.1, and this section defines the one we evaluate. To make memorized and control activations look alike, the natural move acts on the full d_{model}-dimensional difference between them, whether by penalizing it directly as in FitNets (Romero et al., 2015), through a minimax game against a discriminator (Ganin & Lempitsky, 2015), or by the concept-erasure constructions of Section 2. Most of that difference supports the model's other behavior, so constraining all of it costs capability in directions the fitted probe does not read. We therefore compare against an isotropic penalty run on the same recipe (Section 4.4) rather than against LEACE, whose guarantee assumes the concept it erases is fixed.

Probe-geometry alignment (PGA) constrains only the direction the auditor reads. At each depth d we hold a unit-norm readout direction $\hat{w}_d$, refit from the current model every K steps, 25 on Pythia-70M, and penalize only the scalar projection of the memorized-against-control difference onto it:

$$L_{\mathrm{PGA}}(\theta) = \underbrace{\lambda_c \, \mathrm{CE}_{\mathrm{clean}}(M_\theta)}_{\text{capability anchor}} + \frac{\lambda}{N|\mathcal{A}|} \sum_{d \in \mathcal{A}} \sum_{i=1}^{N} \Big(\hat{w}_d^{\top} \big[h_{d,\theta}(P_i^{\mathrm{sec}}) - h_{d,\theta}(P_i^{\mathrm{cln}}) \big]_{\mathrm{pos}=p_{\mathrm{len}}-1} \Big)^2 . \quad (2)$$

Here $\mathcal{A}$ is the set of aligned depths, which excludes the embedding output on the pretrained models, N is the number of memorized and control pairs, and λ_c is 1 in every run except Mistral-7B, where the anchor is weighted at 3. Appendix I gives $\mathcal{A}$ and λ for every run, and Appendix R the claim each supports. Each depth in $\mathcal{A}$ contributes one scalar constraint rather than d_{model} of them, and the direction is refit because it moves under the edit. If the penalty term goes to zero, every pair's projected difference along $\hat{w}_d$ goes with it, and so does the difference of class means along that direction. That removes the first-moment signal a readout along $\hat{w}_d$ uses, and under equal class-conditional covariances it bounds accuracy along that one direction by the majority-class baseline. The bound covers the constrained direction, not a probe free to choose another. Sections 4.4 and 4.5 measure what the objective achieves against an attacker given that freedom.

Equation (2) is the training objective, and every result labeled PGA uses it unless stated otherwise. The same geometry also works without training. Inference PGA removes k probe directions from the residual stream at inference, refitting the probe after each removal, and the zero-shot benchmarks of Section 4.6 use it with $k = 1$ and with $k = 4$, the number of directions needed to bring the probe below 0.55. On GPT-2 Medium the pool supplies far fewer examples than d_{model}, which leaves a single fitted direction under-determined, so rank-k PGA (RK-PGA) projects out, also at inference, the k-dimensional subspace spanned by the top-k eigenvectors of the standardized between-class scatter and variance-difference matrix, with $k = 2$. Multi-position PGA (Section 4.5) moves the trained constraint to mean-pooled representations. The Pythia-70M and Mistral-7B runs train a low-rank adaptation (LoRA) adapter (Hu et al., 2022), and the controlled testbed trains full weights.

4 Results

4.1 The cross-sequence memorization signature

The probe generalizes in all three architectures, from the middle layers onward. Figure 2 plots true LOO accuracy against residual depth for Pythia-70M (Biderman et al., 2023; Black et al., 2022; Gao et al., 2020), GPT-2 Medium (Radford et al., 2019) and Mistral-7B (Jiang et al., 2023) against both nulls, with the transformer-layer mean of Eq. (1) annotated in each panel. The shuffled-label null sits near 0.5 throughout. The pure-distinguishability null is higher and structured, reaching roughly 0.7

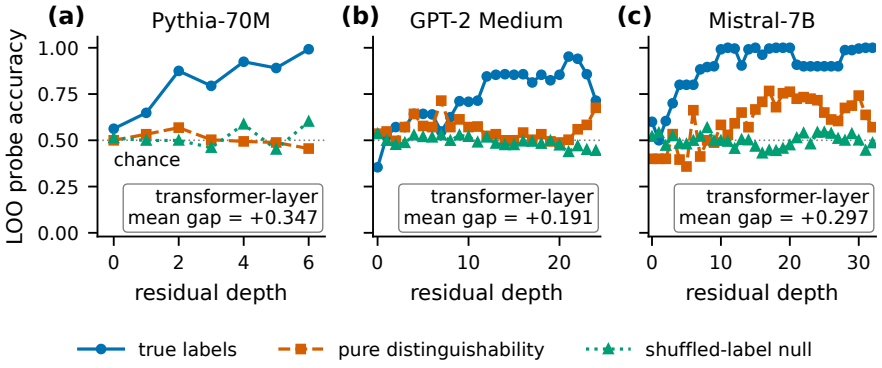

Figure 2: Cross-sequence signature across three architectures.

on GPT-2 Medium and 0.8 on Mistral-7B in mid network, which is the nuisance a study without this control would score as memorization. The true curve clears both.

Transformer-layer mean gaps are $+0.347$ on Pythia-70M, $+0.191$ on GPT-2 Medium and $+0.297$ on Mistral-7B, peaking at $+0.537$, $+0.449$ and $+0.471$. A 100-replicate bootstrap over neutral contexts on GPT-2 Medium gives a 95% interval of $[+0.184, +0.203]$ around that run's own point estimate of $+0.195$. A seven-fold sequence jackknife leaves every held-out gap strictly positive, in $[+0.095, +0.250]$.

Running the identical protocol with the pool fixed at seven sequences on Pythia 70M, 160M, 410M and 1B gives transformer-layer means of $+0.323$, $+0.278$, $+0.334$ and $+0.274$, which sit 8.0, 6.5, 8.2 and 5.1 standard deviations above each model's own shuffled-label null. The peak migrates in relative depth, so the signature is not pinned to a fixed architectural position (Appendix D).

Four controls test architecture, register, vocabulary and content type. A randomly initialized Pythia-70M shows that architecture alone carries part of the effect and pretraining supplies the rest: its transformer-layer mean gap is $+0.188$ against the pretrained $+0.323$, and the two profiles run in opposite directions, random initialization peaking at $+0.327$ in the first layer and falling to -0.042 in the last, while the pretrained model peaks at $+0.643$ at depth four. On GPT-2 Medium the probe reads held-out sequences from all four content clusters above the pure-distinguishability null, so no single register carries the result (Appendix F).

Those pools are selected by likelihood screening. Corpus lookup on the Pile (Gao et al., 2020) instead lifts the pool to 30 sequences and makes the controls constructed rather than authored. The corrected gap peaks at $+0.413$, where the probe's accuracy sits 10.5 standard deviations above a permutation null, with a thirty-fold jackknife spanning only $[+0.362, +0.459]$ (Appendix E).

Finding 1. *A memorization-specific signature generalizes to held-out sequences in every model tested, from a 70M-parameter GPT-NeoX to a 7.24B-parameter Mistral, and is size-stable within a family from 70M to 1B at* $+0.27$ *to* $+0.33$*, five to eight standard deviations above a shuffled-label null. On a verified-membership pool of* 30 *Pile sequences the corrected gap reaches* $+0.413$*, where the probe's accuracy sits* 10.5 *standard deviations above a permutation null.*

4.2 Causal separability of the probe direction

If the direction the probe reads were the direction producing the memorized completion, suppressing it would amount to unlearning by another name. The two come apart. On Pythia-70M, seven-sequence pool, we fit the LOO probe at the peak-gap depth four and intervene at that depth to project its unit-norm weight out of the residual stream, $h' = h - (h \cdot \hat{w})\hat{w}$. The gap collapses at that depth, from $+0.643$ to $+0.161$, partly persists at depth five and partly recovers by depth six, with the transformer-layer mean falling from $+0.323$ to $+0.193$ (Appendix B). The model continues to recite the text. Memorized $\log P/\text{tok}$ falls by between 0.053 and 0.116 nats across the seven sequences, and held-out log-probability moves from -5.138 to -5.163, a degradation of 0.025 nats.

Table 2: Unlearning methods compared on Pythia-70M, unified pipeline. PPL is held-out perplexity, retention is base-model PPL over method PPL, and the probe column is orientation-corrected accuracy at the deepest depth.

Method	Recall supp. (nats)	Held-out PPL	Capability retention	Probe at depth 6
Base model	0.00	191.1	1.000	0.929
GA	62.22	10198.5	0.019	0.929
NPO	18.99	491.1	0.389	0.929
RMU	0.02	285.3	0.670	0.857
IDK	1.05	240.6	0.794	0.929
PGA (ours)	3.25	215.5	0.886	1.000

Downstream layers partially reconstruct the signature, so the structure is re-derived as computation proceeds. That is why a projection at a single depth does not hold and why Section 3.2 constrains a set of depths during training.

Finding 2. *Probe-readable structure and recall-producing structure occupy separable directions. Projecting out the fitted probe direction collapses the local signature while behavioral recall drops by at most* 0.116 *nats per token.*

4.3 Existing unlearning methods

Whether existing unlearning methods reach this structure is cleanest to test on a controlled model. On a four-layer transformer (Vaswani et al., 2017) trained to recall an injected sentence with probability 1.000, causal tracing over attention heads (Meng et al., 2022; Wang et al., 2023) localizes the behavior to one head out of 32, and an edit restricted to that head drives recall to 1.3×10^{-4} while touching 0.76% of the parameters. This is the best case for precision unlearning, since the circuit is localized to one head and the edit is confined to it, and the probe stays at 1.000. Five published methods spanning both families of Ren et al. (2025) behave the same way on that testbed: four reach behavioral erasure, gradient ascent does not, and all five leave the probe at 1.000. Appendix G gives the per-method numbers, and Appendices L, M and O extend the result to eight independently trained secrets, to alignment restricted to the causally identified heads, and to relearning.

Two stronger tests on a nine-secret version of the testbed point the same way (Appendix P). Mass-editing memory in a transformer (MEMIT) (Meng et al., 2023) edits the weights that hold the association rather than training against the output, and takes recall from 0.978 to 8.4×10^{-5} with the probe still at 1.000. Distillation from a teacher that never memorized comes closest to retraining, since it has a model to imitate rather than an objective to descend. It drives the student's recall to 2.6×10^{-5} at a held-out perplexity of 1.405 against 1.403, and the student still reads 0.922 where the teacher's best depth reads 0.661. A model can be trained to reproduce the behavior of one that never memorized and keep a geometry that one does not have.

The same pattern holds at pretrained scale. Table 2 and Figure 3 compare PGA on Pythia-70M, with the natural memorized pool and one scoring protocol, against GA (Yao et al., 2024), NPO (Zhang et al., 2024), RMU (Li et al., 2024) and IDK (Maini et al., 2024). The figure gives probe accuracy by depth under both conventions, recall suppression in nats, and capability retention, and only PGA drives the raw reading below chance.

The probe column does not separate these methods: every one, ours included, leaves the deepest-depth readout between 0.857 and 1.000 while recall suppression ranges over three orders of magnitude.

What the other two columns separate is the price. RMU, the only baseline designed to redirect internal representations, delivers 0.02 nats of suppression at a cost of a third of the model's capability. IDK delivers 1.05 nats at a cost of a fifth. GA and NPO suppress far more by destroying the model, GA at 1.9% retention. PGA delivers 3.25 nats at 88.6% retention, which Section 4.4 examines.

The 1.000 in the PGA row needs reading carefully, because the raw number behind it is the one an audit would quote. The probe column is a fresh leave-one-out logistic regression fit on the aligned model at each depth. At the deepest depth it places every held-out sequence on the wrong side, so the raw reading is 0.000 and an audit quoting it would certify erasure. Alignment did not remove the class information. It reversed the orientation of the readout, and the corrected value says that a reader

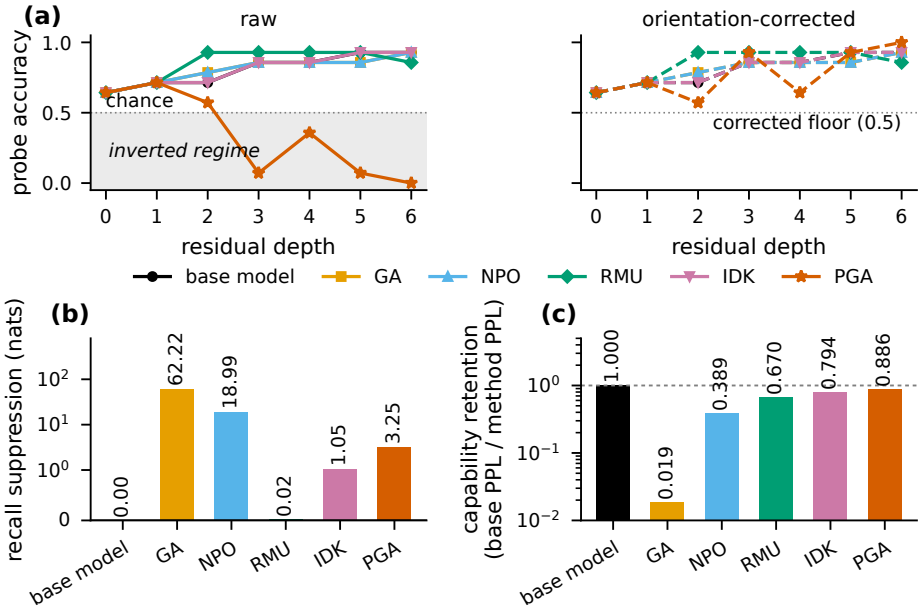

Figure 3: Five unlearning methods on Pythia-70M.

Table 3: Suppression efficiency under a common estimator, Pythia-70M.

Variant	Recall supp. (nats)	Held-out cost (nats)	Nats suppressed per nat lost
PGA, $\lambda = 1.0$	3.25	0.055	**59.40**
Isotropic, $\lambda = 0.001$	4.27	6.107	0.70
Isotropic, $\lambda = 0.01$	5.58	4.665	1.20
Isotropic, $\lambda = 0.1$	4.76	0.532	8.96
Isotropic, $\lambda = 1.0$	5.44	1.781	3.05

who fixes the sign recovers the classes in full. The two readings are complements of one another, and only the corrected one describes what an adversary can do, since the sign flip is free. A stronger audit gives the attacker eight classifier families rather than one, and under the protocol of Section 4.4 the best of the eight reads 0.896 against 0.955 for the base model.

All six rows share one scoring pipeline. The leave-one-out protocol of Section 3.1 reads the same PGA model at 0.857 at the deepest depth, and 0.893 ± 0.036 across four adapter seeds (Appendix I).

4.4 An isotropic ablation

Table 3 measures what the geometric targeting buys by replacing only the geometry. It holds the recipe, layers, seeds, budget and held-out estimator fixed and substitutes an isotropic alignment, a d_{model}-dimensional squared-distance penalty swept over $\lambda \in \{0.001, 0.01, 0.1, 1.0\}$, the last matching PGA's own weight.

The isotropic ablation suppresses recall effectively but not efficiently. It returns 8.96 nats per nat of held-out capability at its most efficient setting and 0.70 at its least. PGA returns 59.4, a factor of 6.6 over the best isotropic setting and 19.5 over the arm trained at the same penalty weight. Against the eight-family probe suite of Section 4.5, read at the last token that both methods align, PGA's highest corrected readout is 0.896, below the ablation's 0.935 and the base model's 0.955, so the directional objective leads on both efficiency and probe suppression.

The margin holds against the best of the four weights tested and against the arm trained at PGA's own weight, which removes the penalty budget as an explanation. With the held-out estimator of the unified pipeline instead, PGA's ratio is 26.9 (Appendix H).

4.5 Adaptive attackers and multi-position alignment

PGA is optimized against a probe family, so we evaluate it against attackers from outside that family. We hand an attacker eight classifier families spanning linear, kernel, instance-based and ensemble methods, free choice of regularization and seed, free negation of the fitted weights, and a pooling scheme PGA never constrains: PGA aligns the last token, and the attacker mean-pools over all positions.

Against last-token PGA on Pythia-70M, a fresh mean-pooled logistic regression reads 0.914, 0.940 and 0.955 at depths three to five, and the strongest of the eight families reaches 0.979. The alignment did not generalize across the position axis, a specific diagnosis that points at its own repair.

Multi-position PGA (MP-PGA) extends the alignment to mean-pooled representations, identical in every other respect. The fresh mean-pooled probe falls to 0.548 at depth three, below the chance expectation of the corrected metric (Appendix C), while recall suppression *rises* from 3.25 to 4.84 nats, at 0.324 nats more held-out cost than last-token alignment. The result is scoped to the linear attacker at mid depths. The strongest non-linear family there still reads 0.789 and the deepest depth returns to 0.929, which is where the construction extends next (Appendix A, Figure 4).

4.6 Capability retention and scale

Removing the probe direction costs little general capability. Inference PGA with $k = 1$ and with $k = 4$ moves no zero-shot benchmark by more than 2.9 percentage points, with mean changes of -0.52 and $+0.19$ points across five tasks. At Mistral-7B scale, PGA suppresses recall by 2.63 nats at a held-out cost of 0.72 nats, which is 3.6 nats per nat. The readout moves with it: corrected accuracy at the five aligned depths falls from 0.988–1.000 to 0.500–0.714, while the best-reading depth, which alignment never targeted, still reads 0.881. Appendices J and R give the per-task numbers and the scale and configuration behind each claim.

5 Discussion

This paper turns representational memorization into something an auditor can measure. One probe, fitted on a handful of memorized texts, reports whether a passage it has never seen is still encoded. The signature holds on every model we tested and is causally separable from the machinery that recites the text, which is why a behavioral test cannot see it and why no behavioral erasure we applied reaches it, from five published unlearning objectives to weight editing to imitation of a model that never memorized.

The same measurement doubles as a training target, and that is where the efficiency comes from. Probe-geometry alignment constrains one scalar per depth, leaves the rest of the residual stream alone, and returns 6.6 times the recall suppression per nat of capability of the best isotropic setting. One result applies beyond this method: a constrained readout can turn over rather than empty, and a reversed readout scores below chance and resembles success, so any erasure claim measured with a probe has to state which way the probe points (Appendix Q).

Limitations and next steps. The instrument measures what an auditor can compute from weights and activations alone, which is whether the target content is still linearly decodable. Three extensions follow directly from that. Establishing pool membership by corpus lookup, as we do for Pythia-70M, applies to any model whose training corpus is public. The protocol needs only new pools to run on content types beyond the software licenses and web boilerplate used here, and on models larger than the 7B we reach. Pairing the audit with extraction measurements would turn a statement about encoding into a risk estimate.

The instrument and the method share one object. What an auditor computes from weights and activations alone is exactly what alignment constrains, so any deployment able to fit a probe can state what its deletion claim is worth, and act on the answer.

REPRODUCIBILITY STATEMENT

The archive accompanying this submission holds the notebook for every experiment with its executed output retained, the verbatim text of every probing pool, the low-rank adapter for the Pythia-70M alignment run, and, for every number reported here, either the result file it was read from or the executed notebook whose retained output prints it. Every figure is generated by released code that reads those result files and writes a vector and a 600 dpi raster version of each panel. No figure contains a hardcoded value, and that code prints the numeric content of every panel on execution, so each number in the text can be traced to a named file. Protocol details are in Appendix C and configurations in Appendix I. The archive will be released publicly.

AI USE STATEMENT

We used generative AI tools to implement analysis and experiment code from our specifications, to suggest additional analyses, from which we chose what to run and judged the results on the evidence, to draft and edit prose, including the description of results, to produce the notebook that regenerates the figures from archived result files, to check the manuscript's numbers against those files, and to format references.

We have not used generative AI tools to run experiments, to generate data, or to produce any reported measurement. Every measurement comes from an executed run whose output file is in the released archive. We verified each reported value against its source file, edited all content, and take responsibility for the paper and its artifacts.

ETHICS STATEMENT

This work studies whether content a model was asked to forget remains encoded in its representations, and proposes a method to suppress an audit signal. Because a method that hides such a signal can be misused, we state the threat model explicitly.

The auditor has white-box access to the weights and activations of a deployed model and a candidate passage, but no training corpus, no pre-unlearning copy of the model and no privileged log. The auditor fits a linear probe on the model's own hidden states from a small pool of sequences the model demonstrably reproduces and a matched pool of controls, and asks whether that probe transfers to the candidate passage. A positive answer is evidence the passage is still linearly encoded. The adversary of Section 4.5 is stronger: it may choose the classifier family, the pooling scheme, the regularization and the seed, and may negate the fitted weights at no cost, though it may not modify the model. Defending against the auditor therefore does not by itself defend against the adversary.

What PGA provides is suppression of a specified audit signal under a stated attacker family, together with a measured reduction in behavioral recall. It is not content deletion, and it is not an extraction-risk guarantee: establishing that would need extraction attacks (Carlini et al., 2021; Nasr et al., 2025) and discoverable-memorization measurement (Carlini et al., 2023), which we do not report. A method that suppresses a probe-readable signal could in principle be used to make a model appear compliant while retaining content, which is why every claim in Section 4 is scoped to the attacker family it was evaluated against, why Appendix R states the scale and variant behind each claim, and why we release the attacker suite alongside the method so that the suppression can be tested rather than assumed. The memorized sequences are software licenses and web boilerplate, and no personal or sensitive data was involved.

REFERENCES

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, 2022.

Nora Belrose, David Schneider-Joseph, Shauli Ravfogel, Ryan Cotterell, Edward Raff, and Stella Biderman. LEACE: Perfect linear concept erasure in closed form. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, et al. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning (ICML)*, 2023.

Stella Biderman, Hailey Schoelkopf, Lintang Sutawika, Leo Gao, Jonathan Tow, Baber Abbasi, Alham Fikri Aji, Pawan Sasanka Ammanamanchi, Sidney Black, Jordan Clive, et al. Lessons from the trenches on reproducible evaluation of language models. *arXiv preprint arXiv:2405.14782*, 2024.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. PIQA: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 7432–7439, 2020.

Sid Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, et al. GPT-NeoX-20B: An open-source autoregressive language model. In *Proceedings of BigScience Episode #5: Workshop on Challenges and Perspectives in Creating Large Language Models*, pp. 95–136, 2022.

Lucas Bourtoule, Varun Chandrasekaran, Christopher A. Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *IEEE Symposium on Security and Privacy (SP)*, 2021.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. Discovering latent knowledge in language models without supervision. In *International Conference on Learning Representations (ICLR)*, 2023. URL https://arxiv.org/abs/2212.03827.

Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In *2015 IEEE Symposium on Security and Privacy*, pp. 463–480. IEEE, 2015.

Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, et al. Extracting training data from large language models. In *USENIX Security Symposium*, pp. 2633–2650, 2021.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramèr, and Chiyuan Zhang. Quantifying memorization across neural language models. In *International Conference on Learning Representations (ICLR)*, 2023.

Haokun Chen, Yueqi Zhang, Yuan Bi, Yao Zhang, Tong Liu, Jinhe Bi, Jian Lan, Jindong Gu, Claudia Grosser, Denis Krompass, et al. Does machine unlearning truly remove knowledge? *arXiv preprint arXiv:2505.23270*, 2025.

Yiwei Chen, Soumyadeep Pal, Yimeng Zhang, Qing Qu, and Sijia Liu. Unlearning isn't invisible: Detecting unlearning traces in LLMs from model outputs. In *International Conference on Learning Representations (ICLR)*, 2026.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 2924–2936, 2019.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try ARC, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. QLoRA: Efficient finetuning of quantized LLMs. *Advances in Neural Information Processing Systems*, 36:10088–10115, 2023.

Ronen Eldan and Mark Russinovich. Who's Harry Potter? approximate unlearning in LLMs. *arXiv preprint arXiv:2310.02238*, 2023.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, et al. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021. URL https://transformer-circuits.pub/2021/framework/index.html.

Chongyu Fan, Jiancheng Liu, Licong Lin, Jinghan Jia, Ruiqi Zhang, Song Mei, and Sijia Liu. Simplicity prevails: Rethinking negative preference optimization for LLM unlearning. *Advances in Neural Information Processing Systems*, 38:1540–1567, 2026.

Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning (ICML)*, 2015.

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.

Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Eternal sunshine of the spotless net: Selective forgetting in deep networks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9301–9309. IEEE, 2020.

Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research (JMLR)*, 13:723–773, 2012.

John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2733–2743, 2019.

John Hewitt and Christopher D. Manning. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2019.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022.

Shengyuan Hu, Yiwei Fu, Zhiwei Steven Wu, and Virginia Smith. Unlearning or obfuscating? jogging the memory of unlearned LLMs via benign relearning. In *International Conference on Learning Representations (ICLR)*, 2025.

Hengrui Jia, Taoran Li, Jonas Guan, and Varun Chandrasekaran. The erasure illusion: Stress-testing the generalization of LLM forgetting evaluation. *arXiv preprint arXiv:2512.19025*, 2025.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, et al. Mistral 7B. *arXiv preprint arXiv:2310.06825*, 2023.

Nathaniel Li, Alexander Pan, Anjali Gopal, Summer Yue, Daniel Berrios, Alice Gatti, Justin D. Li, Ann-Kathrin Dombrowski, et al. The WMDP benchmark: Measuring and reducing malicious use with unlearning. In *International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pp. 28525–28550, 2024.

Aengus Lynch, Phillip Guo, Aidan Ewart, Stephen Casper, and Dylan Hadfield-Menell. Eight methods to evaluate robust unlearning in LLMs. *arXiv preprint arXiv:2402.16835*, 2024.

Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary C. Lipton, and J. Zico Kolter. TOFU: A task of fictitious unlearning for LLMs. In *First Conference on Language Modeling (COLM)*, 2024.

Samuel Marks and Max Tegmark. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. In *First Conference on Language Modeling (COLM)*, 2024.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pp. 17359–17372, 2022.

Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. Mass-editing memory in a transformer. In *International Conference on Learning Representations (ICLR)*, 2023.

Milad Nasr, Javier Rando, Nicholas Carlini, Jonathan Hayase, Matthew Jagielski, A. Feder Cooper, Daphne Ippolito, Christopher Choquette-Choo, Florian Tramèr, and Katherine Lee. Scalable extraction of training data from aligned, production language models. In *International Conference on Learning Representations (ICLR)*, pp. 82363–82435, 2025.

Markus Ojala and Gemma C. Garriga. Permutation tests for studying classifier performance. *Journal of Machine Learning Research*, 11(6), 2010.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 2019.

Inioluwa Deborah Raji, Andrew Smart, Rebecca N. White, Margaret Mitchell, Timnit Gebru, Ben Hutchinson, Jamila Smith-Loud, Daniel Theron, and Parker Barnes. Closing the AI accountability gap: Defining an end-to-end framework for internal algorithmic auditing. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pp. 33–44, 2020.

Shauli Ravfogel, Yanai Elazar, Hila Gonen, Michael Twiton, and Yoav Goldberg. Null it out: Guarding protected attributes by iterative nullspace projection. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020.

Shauli Ravfogel, Michael Twiton, Yoav Goldberg, and Ryan Cotterell. Linear adversarial concept erasure. In *International Conference on Machine Learning (ICML)*, 2022.

Jie Ren, Yue Xing, Yingqian Cui, Charu C. Aggarwal, and Hui Liu. SoK: Machine unlearning for large language models. *arXiv preprint arXiv:2506.09227*, 2025.

Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. FitNets: Hints for thin deep nets. In *International Conference on Learning Representations (ICLR)*, 2015.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. WinoGrande: An adversarial Winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. Detecting pretraining data from large language models. In *International Conference on Learning Representations (ICLR)*, 2024.

Weijia Shi, Jaechan Lee, Yangsibo Huang, Sadhika Malladi, Jieyu Zhao, Ari Holtzman, Daogao Liu, Luke Zettlemoyer, et al. MUSE: Machine unlearning six-way evaluation for language models. In *International Conference on Learning Representations (ICLR)*, 2025.

Kushal Tirumala, Aram H. Markosyan, Luke Zettlemoyer, and Armen Aghajanyan. Memorization without overfitting: Analyzing the training dynamics of large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

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.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.

Changsheng Wang, Yihua Zhang, Jinghan Jia, Parikshit Ram, Dennis Wei, Yuguang Yao, Soumyadeep Pal, Nathalie Baracaldo, et al. Invariance makes LLM unlearning resilient even to unanticipated downstream fine-tuning. In *International Conference on Machine Learning (ICML)*, 2025.

Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: A circuit for indirect object identification in GPT-2 small. In *International Conference on Learning Representations (ICLR)*, 2023.

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.

Yuanshun Yao, Xiaojun Xu, and Yang Liu. Large language model unlearning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4791–4800, 2019.

Dawen Zhang, Pamela Finckenberg-Broman, Thong Hoang, Shidong Pan, Zhenchang Xing, Mark Staples, and Xiwei Xu. Right to be forgotten in the era of large language models: Implications, challenges, and solutions. *AI and Ethics*, 5(3):2445–2454, 2025a. doi: 10.1007/s43681-024-00573-9.

Ruiqi Zhang, Licong Lin, Yu Bai, and Song Mei. Negative preference optimization: From catastrophic collapse to effective unlearning. In *First Conference on Language Modeling (COLM)*, 2024.

Zhiwei Zhang, Fali Wang, Xiaomin Li, Zongyu Wu, Xianfeng Tang, Hui Liu, Qi He, Wenpeng Yin, et al. Catastrophic failure of LLM unlearning via quantization. In *International Conference on Learning Representations (ICLR)*, 2025b.

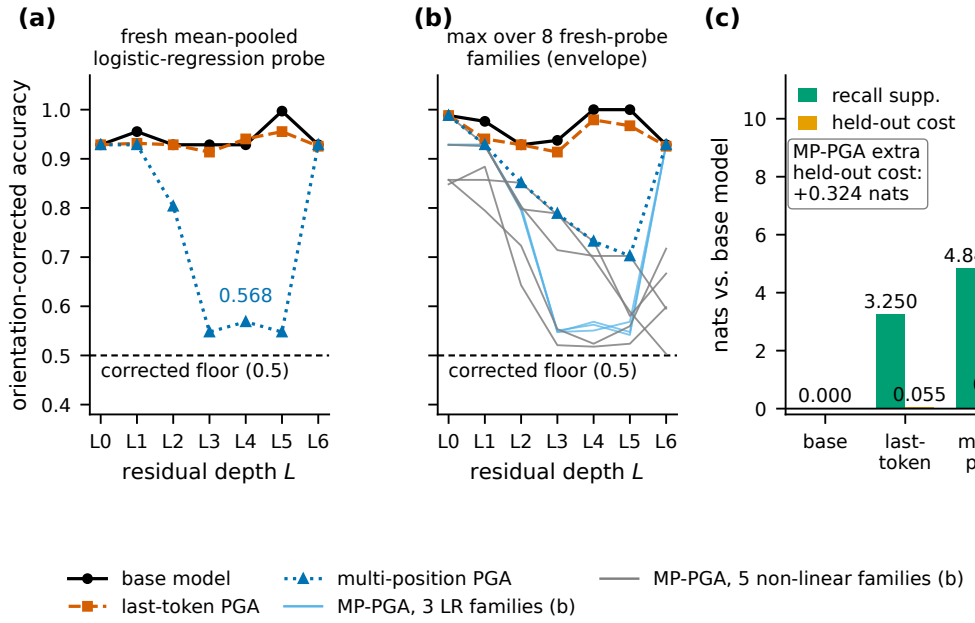

Figure 4: Multi-position alignment against a mean-pooled attacker. (a) A fresh mean-pooled logistic regression. (b) The maximum over eight fresh-probe families. (c) Recall suppression and held-out cost for both variants.

A Multi-position alignment

Last-token PGA constrains the readout at one position. The attacker of Section 4.5 pools over positions, and that is the axis along which the alignment failed to generalize. Multi-position PGA extends the same one-scalar-per-depth constraint to mean-pooled representations, leaving the loss weights, the refit schedule, the layers, the seeds and the capability anchor unchanged.

The repair is specific to the axis the diagnosis named. Against a fresh mean-pooled logistic regression the readout falls to 0.548, 0.568 and 0.548 at depths three to five, below the chance expectation of the corrected metric, while recall suppression rises from 3.25 to 4.84 nats. Because the objective and the attacker now share a pooling scheme, this is confirmation that the account localizes the failure correctly rather than an independent prediction. The strongest non-linear family at those depths reads 0.789, 0.732 and 0.702, the two multilayer perceptrons fall with the linear probes, and the final depth returns to 0.929.

B Probe-direction ablation

Figure 5 shows the intervention of Section 4.2. Panel (a) gives the memorization-specific gap by depth before and after the fitted probe direction is projected out of the residual stream at depth four. Panel (b) gives the change in $\log P/\mathrm{tok}$ that the same intervention causes for each memorized sequence and for held-out text. No memorized sequence moves by more than 0.116 nats.

C Leave-one-out protocol details

The memorization-specific gap of Eq. (1) subtracts raw leave-one-out accuracies, while the readout numbers reported in Section 4 are orientation-corrected. The two conventions answer different questions. The corrected metric asks what an adversary can recover, and the sign flip is free to an adversary. A null, by contrast, is centered on chance, and $\max(a, 1-a)$ applied to a chance-centered statistic has expectation strictly above 0.5: simulating an uninformative probe gives a corrected mean of 0.656 at seven folds and 0.572 at thirty, against a raw mean of 0.500 in both cases. Correcting the

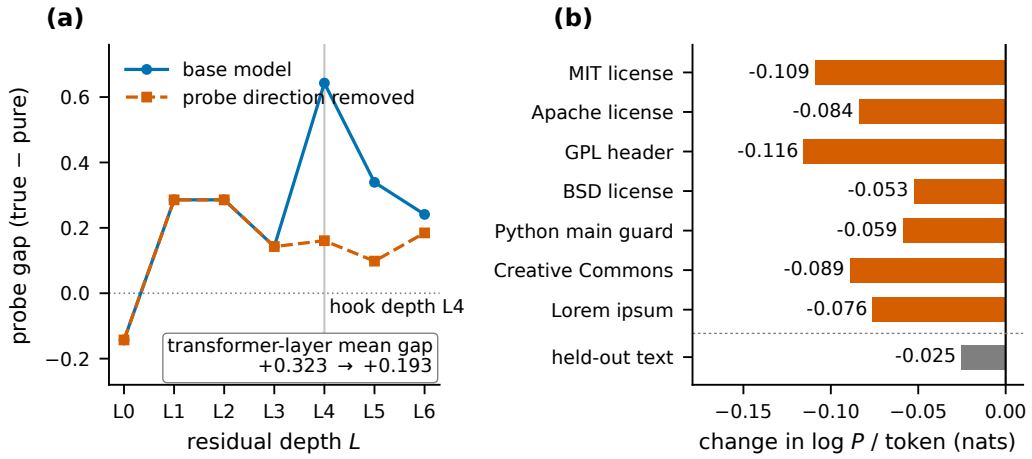

Figure 5: Ablating the probe direction on Pythia-70M.

pure-distinguishability null would therefore subtract that sampling noise from every gap as though it were nuisance separability, and the subtraction would be largest exactly where the pools are smallest. The verified-membership pool of Appendix E is corrected on both terms, which makes its gap a conservative estimate.

Screening thresholds are $(\theta_M, \theta_C) = (-2.0, -3.0)$ for Pythia-70M and $(-1.5, -2.5)$ for GPT-2 Medium, which share a pool definition. Mistral-7B uses a pre-screened set with POOL_D re-screened at $\log P/\mathrm{tok} < -2.5$. Screening on both thresholds drops the candidate set from nine keys to seven for Pythia-70M and GPT-2 Medium. The Mistral-7B pool contains five pre-validated pairs.

For each key j, each context $b \in \mathcal{B}$ and each class $X \in \{\mathrm{POOL}_M, \mathrm{POOL}_C, \mathrm{POOL}_D\}$ we extract the last-token residual-stream activation of the concatenation $b \| X[j]$ at every depth, giving three tensors of shape $(|\mathcal{K}_{\mathrm{valid}}|, |\mathcal{B}|, D{+}1, d_{\mathrm{model}})$. POOL_D[`mit_license`] begins "Authorization is hereby withheld, universally and permanently". The verbatim text of every pool entry is released with the archive, where POOL_D is labeled pool B and $\mathcal{B}$ the neutral contexts.

Table 4 collects the notation used throughout.

Table 4: Notation.

Symbol	Meaning
M_θ	Language model with parameters θ
P_i^{sec}, P_i^{cln}	Memorized prefix i and its register-matched control
POOL_M	Memorized sequences passing the screening threshold
POOL_C	Register-matched paraphrase controls, one per member of POOL_M
POOL_D	Second, further paraphrased control per key
$\mathcal{B}$	Pool of 24 neutral context prefixes
$h_{d,\theta}$	Residual stream at depth d under parameters θ, with $d = 0$ the embedding output
$\hat{w}_d$	Unit-norm probe readout direction at depth d
g_d	Memorization-specific gap at depth d, Eq. (1)
$\mathrm{acc}^{\pm}$	Orientation-corrected accuracy $\max(\mathrm{acc}, 1 - \mathrm{acc})$
$\mathcal{A}$	Set of aligned depths, excluding the embedding on pretrained models
k	Number of constrained directions per depth, 1 unless stated
λ, λ_c	Alignment weight and capability weight in Eq. (2)

Two Pythia-70M pools appear in this paper. The eight-sequence pool screens on the memorized threshold alone and so keeps `shebang_python`, whose control misses θ_C at -2.675 and is dropped from the seven-sequence pool. It is evaluated over five probe seeds, giving a mean gap of $+0.347$ with a peak at depth six. The seven-sequence screened pool, held fixed across model sizes so that Appendix D compares like with like, gives $+0.323$ and peaks at depth four with a gap of $+0.643$. The

three-architecture comparison of Section 4.1, Table 5 and Appendix K use the eight-sequence pool, Appendix E uses the verified pool, and every other Pythia number belongs to the seven-sequence pool. Table 5 collects the per-architecture numbers. Five probe seeds move the gap by less than 0.001, which reflects the determinism of the solver rather than data-level stability, so stability claims rest on resampling instead.

Table 5: Memorization-specific gap by architecture.

Model	Params	Depths	Pool	Mean gap	Peak gap (depth)
Pythia-70M	70M	7	8	+0.347	+0.537 (6)
GPT-2 Medium	355M	25	7	+0.191	+0.449 (21)
Mistral-7B	7.24B	33	5	+0.297	+0.471 (11)

D SIZE SCALING WITHIN THE PYTHIA FAMILY

Figure 6 reports the identical protocol with the pool fixed at seven sequences on Pythia 70M, 160M, 410M and 1B. Transformer-layer means are +0.323, +0.278, +0.334 and +0.274, against shuffled-label nulls whose means sit within 0.014 of chance with standard deviations between 0.041 and 0.053, taken over transformer layers with one degree of freedom removed. That places each model 8.0, 6.5, 8.2 and 5.1 standard deviations above its own null. The peak sits at 0.67, 0.92, 0.38 and 0.69 of network depth respectively.

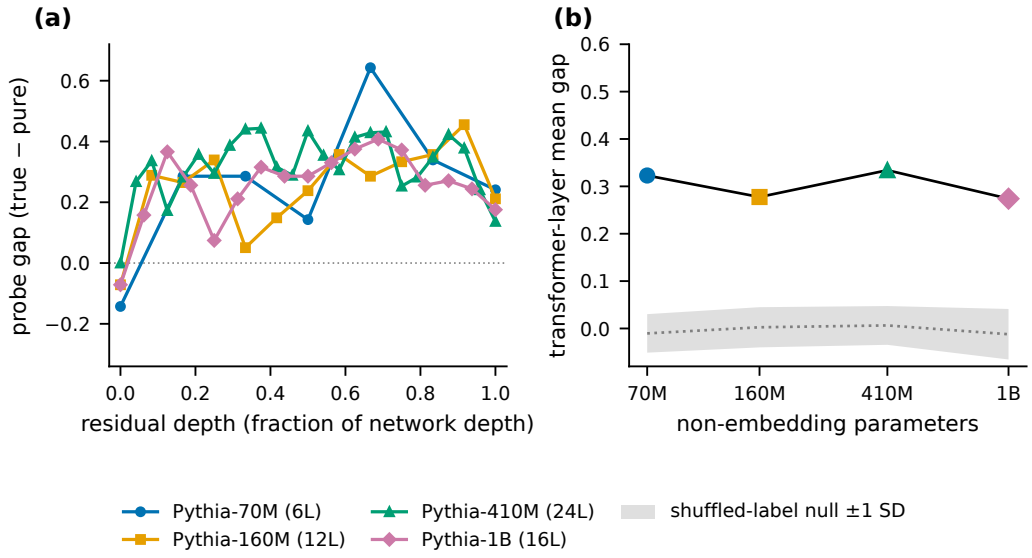

Figure 6: Size scaling within the Pythia family. (a) Gap against relative depth for four sizes on a pool fixed at seven sequences. (b) Transformer-layer mean gap against parameter count with the shuffled-null band.

E VERIFIED-MEMBERSHIP POOL

Pool membership in the main protocol is established by likelihood screening, which selects for verbatim-reproduction risk and not for membership. Pythia was trained on the Pile (Gao et al., 2020), so drawing the pool from the Pile itself replaces the screen with corpus lookup.

We sample 4,000 Pile documents and score each under the base model two ways: log probability per token, and a teacher-forced reproduction score, the fraction of continuation positions where the greedy token matches the true token. Ranking is by reproduction, because likelihood is confounded

by sequence entropy and low-entropy boilerplate scores high whether or not the model stored it. The top 30 form POOL_M and two disjoint bottom blocks form the controls, drawn subset by subset so that all three pools have identical Pile-subset composition. The control set is therefore constructed rather than authored.

The memorized block reproduces 0.980 ± 0.014 of continuation tokens against 0.353 ± 0.046 for its matched controls. The gap peaks at $+0.413$ at depth three. At that depth the true LOO accuracy of 0.965 sits 10.5 standard deviations above a 200-permutation null whose mean is 0.556 (Figure 7). Every one of the thirty leave-one-out values is positive, ranging from $+0.362$ to $+0.459$ with mean $+0.410$.

Twenty-eight of the thirty sequences are GitHub documents and two are PubMed Central. This pool therefore buys verified membership and pool size, not content diversity. Diversity is what the content-type control in Appendix F tests.

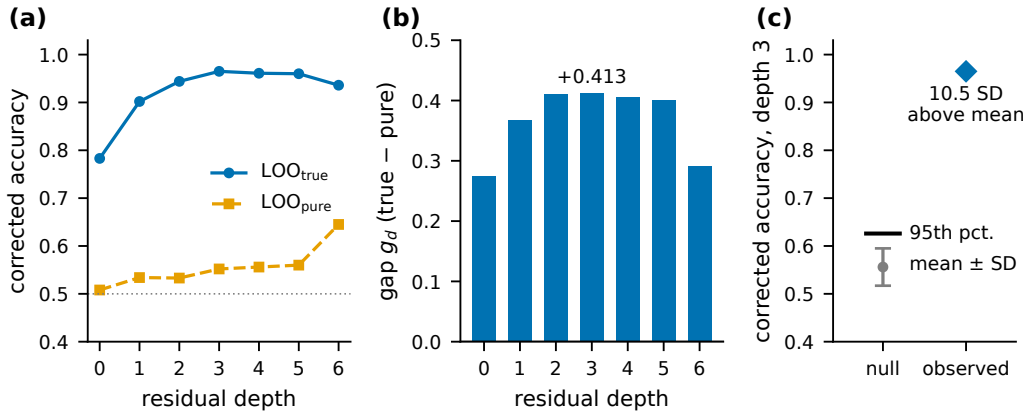

Figure 7: Verified-membership pool from the Pile, Pythia-70M, $n = 30$. (a) Orientation-corrected LOO accuracy by depth. (b) Memorization-specific gap. (c) Observed accuracy at the peak depth against the permutation null.

F Controls

A randomly initialized Pythia-70M controls for architecture, on the seven-sequence pool. It gives a mean transformer-layer gap of $+0.188$ against the pretrained $+0.323$, so architecture alone accounts for roughly three fifths of the average effect and the control is not a null. What separates the two is where the gap sits. Random initialization peaks at $+0.327$ at depth one and decays to -0.042 at depth six. The pretrained model runs the other way, reaching $+0.643$ at depth four and holding $+0.241$ at depth six. Depth structure is therefore a product of pretraining rather than of architecture, consistent with memorization emerging over the course of training (Tirumala et al., 2022), and it is the deep-layer reading that every claim in this paper rests on.

A register control applies the depth-21 GPT-2 Medium probe to 19 unmemorized but register-matched legal passages, where it calls only 14.9% memorized, against 93.9% at the embedding layer. The code control pool is classified as memorized at several depths including 21. Several of those code snippets themselves pass the memorization screening threshold, at $\log P/\mathrm{tok}$ between -1.12 and -1.41, so the depth-21 classification is a correct detection rather than a register artifact.

A vocabulary-matched control on Pythia-70M builds 12 memorized and control continuations sharing a prefix and matched on token distribution, which removes the surface-token route. Held-out probe accuracy stays between 0.590 and 0.727 across transformer depths, and between 0.640 and 0.727 from depth two onward.

Content type is the fourth control. At the peak depth on GPT-2 Medium the probe classifies held-out sequences from all four content clusters above the pure-distinguishability null: legal license text at 0.917 against a null of 0.500, code at 1.000 against 0.500, web boilerplate at 1.000 against 0.521,

and pseudo-Latin at 1.000 against 0.500. Legal text contributes four of the seven sequences and the other three clusters one each.

G THE CONTROLLED TESTBED

The pipeline proceeds in three phases. Phase 1 trains a four-layer, eight-head transformer (Vaswani et al., 2017) with $810{,}496$ parameters, vocabulary size 43 and sequence length 64 on character-level text at an injection density of 16.67%, until $P(S \mid \mathrm{prefix}(S)) = 1.000$ with held-out perplexity 1.39. Phase 2 computes the normalized causal effect (NCE) per head,

$$\mathrm{NCE}(l,h) = \frac{P^{(l,h)}_{\mathrm{patch}} - P_{\mathrm{corrupt}}}{P_{\mathrm{clean}} - P_{\mathrm{corrupt}}}, \tag{3}$$

which reaches 1.000 at head L0H7. Phase 3 freezes all parameters except $\mathbf{W}_q$, $\mathbf{W}_k$, $\mathbf{W}_v$ for heads in the target set and optimizes two objectives independently, $\mathcal{L}_{QK} = \mathcal{L}_{\mathrm{LM}} + \gamma \cdot \mathrm{MMD}^2$ and $\mathcal{L}_V = \mathcal{L}_{\mathrm{LM}} + \alpha \cdot \mathcal{L}_{\mathrm{recall}}$, where MMD is the maximum mean discrepancy (Gretton et al., 2012). Separating the objectives avoids the gradient interference observed under joint optimization. The value term is the component that does the work. Optimizing $\mathcal{L}_V$ alone takes the secret to 3.5×10^{-6} at a held-out perplexity of 1.176 against 1.39 for the base model, and the split objective reaches 1.3×10^{-4}. Both leave the cross-sequence probe at 1.000, which is the reading that matters here.

Two further controls locate the reason. Matched-prefix decoys, contrasting `...Orion is 88492` with `...Orion is 12345` so that every token before the completion is shared, leave probe accuracy at 1.000, so the probe is not reading the completion tokens. Zeroing the embedding rows for the secret tokens moves both together, reducing the secret probability by more than five orders of magnitude and driving the probe to 0.500 ± 0.000. Recall and probe readability can be destroyed at once, and this is the only intervention we found that does it.

Table 6 extends the test to five published methods on the same testbed and probe protocol, with the split-objective edit of this paper listed alongside them. Gradient ascent is the one method that does not reach behavioral erasure here, retaining the secret at probability 1.0 at a perplexity of 382.2.

Table 6: Unlearning methods on the controlled testbed.

Method	Type	$P(\mathrm{secret})$	PPL	Probe
GA	Divergence	1.0×10^{0}	382.2	1.000
NPO	Divergence	5.5×10^{-13}	4.8	1.000
SimNPO	Divergence	1.2×10^{-4}	3.9	1.000
RMU	Representation	2.3×10^{-5}	403.3	1.000
IDK	Rejection	4.2×10^{-8}	4.7	1.000
Split edit (ours)	Mechanistic	1.3×10^{-4}	336.7	1.000

GA, NPO, SimNPO (Fan et al., 2026), RMU (Li et al., 2024) and IDK (Maini et al., 2024) are published methods and fall in both families of Ren et al. (2025). The split edit is the one introduced in Section 4.3, listed for comparison and labeled mechanistic. It modifies 0.76% of parameters and RMU 26.1%. The other four modify all of them.

The testbed results of Appendices I, M, P and Q use a second model of the same architecture, trained on nine memorized and nine control prefixes so that the leave-one-out probe has held-out prefixes to classify. Its held-out perplexity is 1.403.

H INTERPRETATION OF THE ISOTROPIC MARGIN

Two properties of the sweep in Table 3 bear on how the margin should be read.

The isotropic held-out cost is not monotone in λ. It falls from 6.107 to 0.532 nats as the weight rises to 0.1, then rises again to 1.781 at $\lambda = 1.0$. A reader extrapolating from the first three weights would expect the isotropic arm to become competitive at PGA's own weight. It does not: at $\lambda = 1.0$ the

isotropic arm returns 3.05 nats per nat against PGA's 59.4, so a larger penalty does not make isotropic alignment more efficient.

Figure 8 plots the trade-off behind Table 3, with the isotropic arm named AAE in the archive. Panel (a) gives recall suppression and held-out capability cost, both in nats, for PGA and for the isotropic arm at each of its four weights, with the ratio of the two above each pair. Panel (b) gives corrected probe accuracy by depth for the base model before alignment, for PGA, and for the recall-matched isotropic arm at $\lambda = 0.001$, against the permutation-null 95th percentile. Lower is better in panel (b), because it means the probe recovers less.

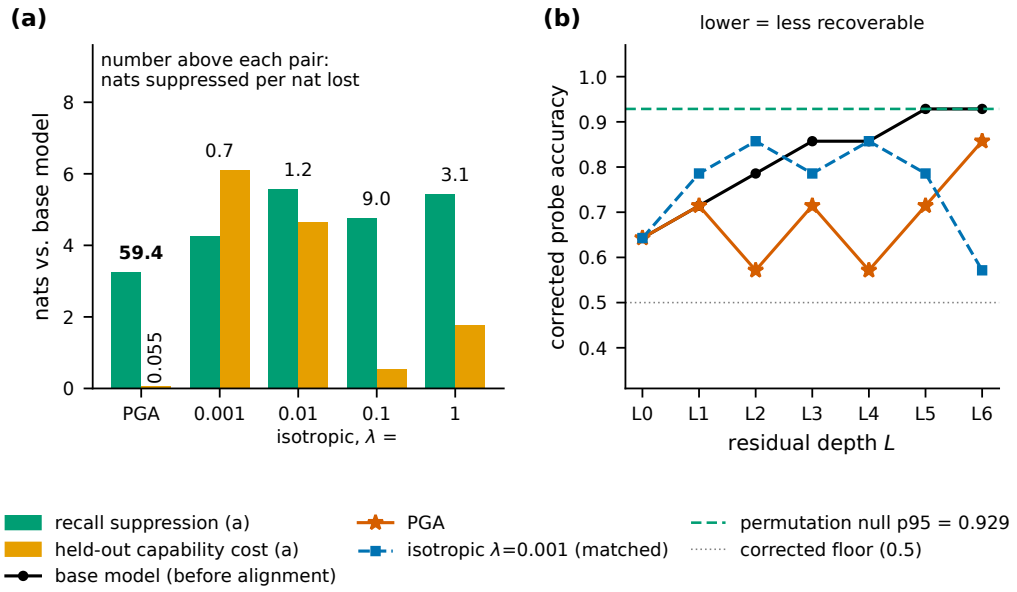

Figure 8: PGA against isotropic alignment on Pythia-70M.

The efficiency ratio depends on which held-out set measures capability. Table 3 holds the estimator fixed across arms, which is the comparison the claim rests on, and there PGA costs 0.055 nats. Measured instead through the unified pipeline of Section 4.3, which uses a different held-out set, the cost is 0.121 nats and the ratio is 26.9 rather than 59.4. The direction of the advantage is therefore robust to the choice of estimator and its magnitude is not, which is why the two numbers are reported separately rather than averaged.

I ALIGNMENT CONFIGURATIONS

The paper uses four variants of the same geometry. **PGA** trains the model with the objective of Equation (2), which constrains one probe direction per aligned depth at the final token. **Multi-position PGA (MP-PGA)** applies the same objective to mean-pooled representations instead of the final token and is otherwise identical. **Inference PGA** involves no training. It removes k probe directions from the residual stream at inference and refits the probe after each removal. **Rank-k PGA (RK-PGA)** also works at inference, and removes the k-dimensional subspace spanned by the top-k eigenvectors of the standardized between-class scatter and variance-difference matrix rather than directions found by repeated probe fits. It is used on GPT-2 Medium, where the pool supplies far fewer examples than $d_{\text{model}} = 1024$ and a single fitted direction is under-determined. In the archive this variant is named MD-PGA. Table 7 lists every configuration used in this paper.

Pythia-70M runs use 200 epochs. Mistral-7B uses four-bit quantization with LoRA (Dettmers et al., 2023), a capability weight of 3.0 and 80 epochs. The readout is refit every 25 steps on Pythia-70M and every 10 on Mistral-7B, one step being one pass over the pool.

Table 7: Alignment configurations.

Variant	Model	Depths	Training	λ
PGA	Pythia-70M	1–6	LoRA r=16	1.0
Inference PGA (k=1)	Pythia-70M	5–6	none	n/a
Inference PGA (k=4)	Pythia-70M	4–6	none	n/a
MP-PGA	Pythia-70M	1–6	LoRA r=16	1.0
RK-PGA (k=2)	GPT-2 Medium	21	none	n/a
Inference PGA (k=5–30)	Testbed	0–4	none	n/a
PGA	Mistral-7B	16, 20, 24, 28, 32	LoRA r=8	0.05
PGA	Testbed	0–4	full weights	0.1

Figure 9 reports corrected probe accuracy by depth, before and after alignment, at all four scales, the three pretrained models plus the controlled testbed. After alignment the lowest depth reads 0.550 on the controlled testbed, 0.571 on Pythia-70M and 0.500 on Mistral-7B, while the depth the probe reads best still reads 0.811, 0.857 and 0.881. On GPT-2 Medium, where the run stores a single evaluated depth, RK-PGA moves depth 21 from 1.000 to 0.939. Across four LoRA seeds on Pythia-70M the corrected depth-six reading is 0.893 ± 0.036. In a separate robustness audit over six fresh probe variants, four linear and two multilayer perceptrons, the corrected worst case is 0.85 on the controlled testbed and 0.93 at Pythia-70M, which sits at the permutation null's 95th percentile of 0.929 for the depth-maximized logistic-regression statistic. Both are recorded to two decimal places in the audit file.

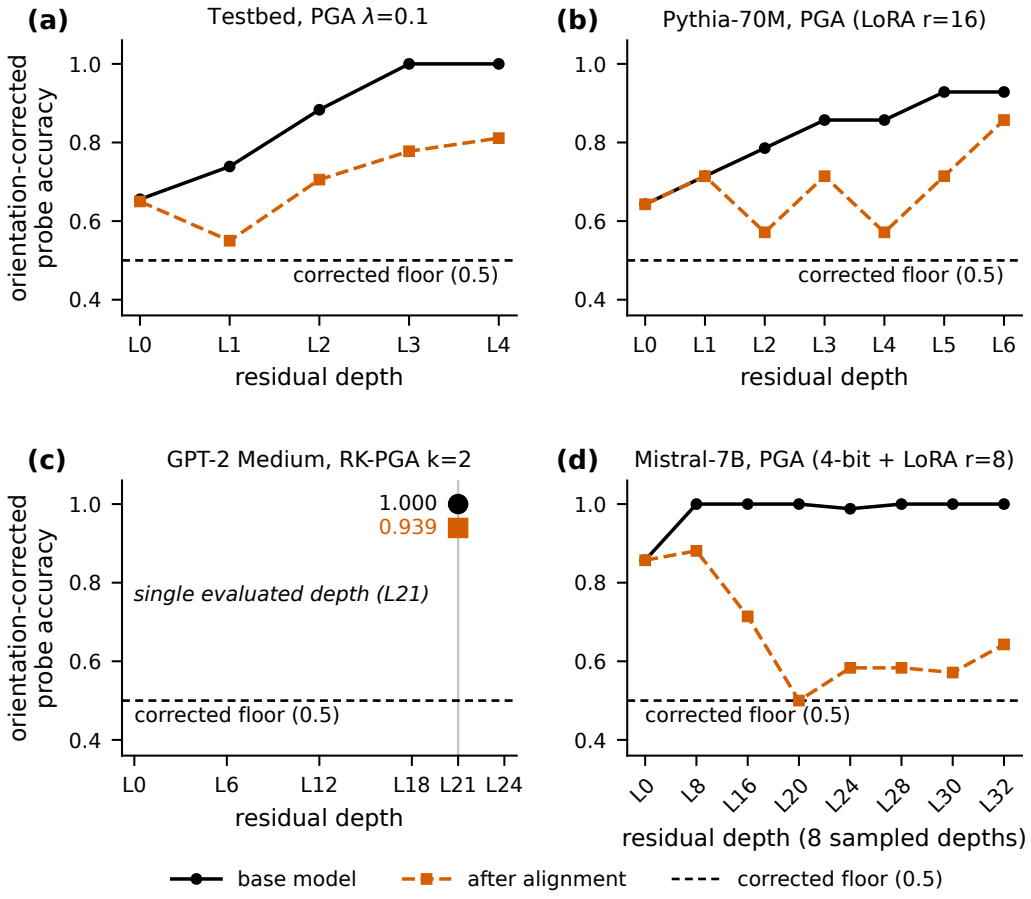

Figure 9: Corrected probe accuracy at four scales, base model against the aligned model, by residual depth. Dashed line marks the floor of the corrected scale.

J CAPABILITY BENCHMARKS

Figure 10 reports five zero-shot benchmarks: HellaSwag (Zellers et al., 2019), PIQA (Bisk et al., 2020), BoolQ (Clark et al., 2019), ARC-Easy (Clark et al., 2018) and WinoGrande (Sakaguchi et al., 2021), all run with the Language Model Evaluation Harness (Biderman et al., 2024). The intervention is inference PGA with $k = 4$, applied at depths four to six (layers three to five in the figure). Panel (b) shows the targeted probe falling as directions are removed, reaching 0.48, below the 0.55 tolerance, at the fourth. Table 8 gives the per-task numbers for $k = 4$ and for $k = 1$, which removes the single probe direction at depths five and six. Neither moves a task by more than 2.9 points in either direction, the two largest movements under $k = 4$ run opposite each other, and the $k = 1$ mean of -0.52 points is within the spread of the tasks themselves.

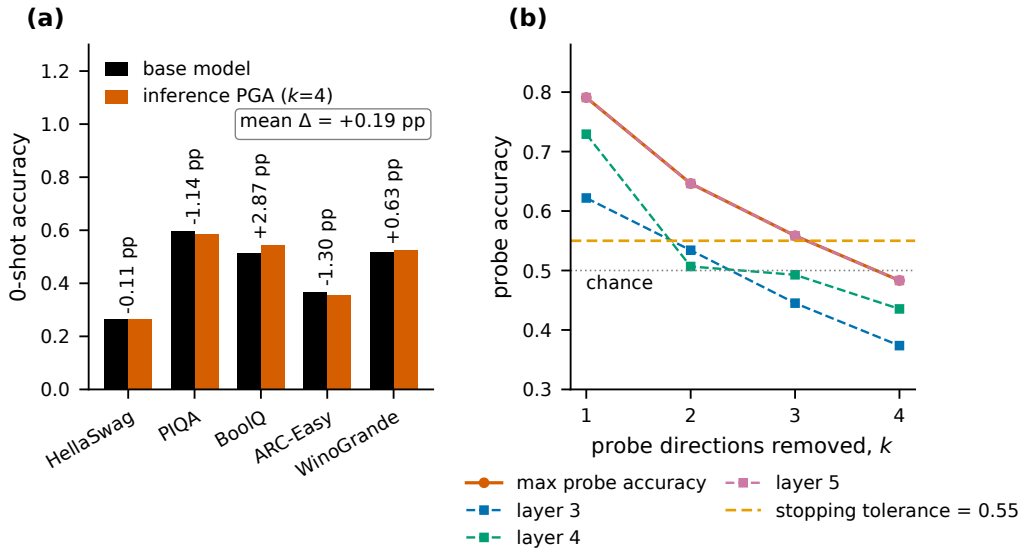

Figure 10: Zero-shot capability under inference PGA, $k = 4$. (a) Per-task accuracy. (b) Targeted probe accuracy by number of directions removed.

Table 8: Zero-shot accuracy, Pythia-70M.

Task	Base	Δ, $k = 1$	Δ, $k = 4$
HellaSwag	26.62	-0.04	-0.11
PIQA	59.79	-1.14	-1.14
BoolQ	51.56	-2.17	$+2.87$
ARC-Easy	36.83	-0.51	-1.30
WinoGrande	51.93	$+1.26$	$+0.63$
Mean		-0.52	$+0.19$

K PER-DEPTH LEAVE-ONE-OUT RESULTS

Table 9 lists the screening log-probabilities for the eight-sequence Pythia-70M pool. One candidate, a Wikipedia passage about Python, fell below the screening threshold at -3.754 and was excluded.

Table 10 gives the three quantities behind Eq. (1) at every depth on that pool. The true LOO accuracy rises with depth while both nulls stay within 0.13 of chance at every depth, so the gap is carried by the true condition alone. Averaging the gap over the six transformer layers gives $+0.347$, and the peak is $+0.537$ at depth six. Both figures are identical across the five probe seeds to four decimal places (Appendix C).

Table 9: Screened pool, Pythia-70M.

Sequence	$\log P/\mathrm{tok}$	Cluster
MIT license	-0.624	legal
Apache license	-0.499	legal
GPL header	-0.467	legal
BSD license	-1.987	legal
Python shebang	-1.565	code
Python main guard	-1.257	code
Creative Commons	-0.935	web
Lorem ipsum	-0.931	placeholder
Wikipedia Python	-3.754	excluded

Table 10: Leave-one-out accuracy by depth, Pythia-70M.

Depth	True	Pure	Shuffled	Gap
0 (embedding)	0.563	0.500	0.500	+0.063
1	0.648	0.533	0.471	+0.116
2	0.875	0.568	0.375	+0.307
3	0.794	0.503	0.477	+0.291
4	0.924	0.494	0.526	+0.430
5	0.891	0.488	0.474	+0.403
6	0.992	0.455	0.549	+0.537

At depth six, held-out accuracy is 1.000 for six of the eight sequences, with the GPL header at 0.979 and the Creative Commons text at 0.958. No sequence carries the result on its own.

L Generalization across secrets

One memorized string is a thin base for a claim about memorization. We therefore ran the controlled testbed of Appendix G independently on eight secrets spanning five structural types. Each secret received its own freshly initialized model, its own causal tracing run and its own unlearning pass, with no component shared across secrets. Table 11 reports the outcome.

Table 11: Eight secrets on the controlled testbed.

Secret	Type	NCE	$\lvert\mathcal{H}\rvert$	$P(\mathrm{sec})$	Probe
88492	numeric	0.654	1	0.995	1.000
31756	numeric	0.764	11	0.019	1.000
XK9-42	alphanumeric	0.881	5	0.243	1.000
Dr. Chen	name	0.775	11	0.008	1.000
03-17	date	0.658	17	2×10^{-4}	1.000
Maple St.	location	0.534	2	0.999	1.000
2947	numeric	0.740	3	0.142	1.000
R7-119	alphanumeric	0.830	9	3×10^{-5}	1.000

Behavioral erasure succeeds for four of the eight, and the four successes are exactly the four with nine or more heads in the target set, against one to five heads for the four failures. We report that separation as a correlation, with training length and the recall weight varied together, so the mechanism behind it is open. The column that matters here is the last one. The probe reads 1.000 after unlearning on all eight, whether or not the behavior was erased.

This probe is saturated by construction, because each testbed model holds a single secret and the memorized class therefore contains one point. The result is a within-model statement that the dissociation does not depend on the choice of secret. It is not cross-sequence evidence, which comes from the pretrained models of Section 4.1.

M ALIGNMENT RESTRICTED TO CAUSALLY IDENTIFIED HEADS

PGA constrains a readout of the full residual stream at each aligned depth. A reader may reasonably ask whether that is necessary, given that causal tracing localizes recall to a few heads. We tested that narrower option directly. Causally localized PGA (CLPA) computes the NCE per attention head on each of the nine memorized prefixes and aligns only at the outputs of the three heads with the largest effect, three of the thirty-two. Table 12 compares the two.

Table 12: Head-local against full-residual alignment.

Configuration	$\min P(\text{sec})$	PPL	Max probe
Base model	9.8×10^{-1}	1.403	1.000
CLPA, $\lambda = 0.1$	3.2×10^{-5}	1.380	0.894
CLPA, $\lambda = 1.0$	3.0×10^{-5}	1.380	0.856
CLPA, $\lambda = 10.0$	2.7×10^{-5}	1.380	0.833
PGA, $\lambda = 0.1$	6.7×10^{-4}	1.420	0.811
PGA, $\lambda = 1.0$	5.9×10^{-5}	1.523	0.778
PGA, $\lambda = 10.0$	8.1×10^{-6}	1.634	0.778

CLPA does better on the two behavioral columns. It suppresses recall as deeply as PGA and it costs nothing in perplexity, sitting fractionally below the unedited model. It moves the probe less than full-residual alignment at every penalty weight, leaving it between 0.833 and 0.894 against 0.778 to 0.811 for PGA. The alignment loss converges to about 5×10^{-3} at all three weights, so the constraint is met at the three heads it was applied to. The other twenty-nine are left unconstrained, and the probe still reads the signal.

This is the dissociation at finer grain. The heads that carry recall are not the heads that carry the readable signature, and an erasure method inherits the geometry of whatever it constrains. It is also the ablation that justifies the cost of PGA, since the perplexity PGA spends buys the lower probe reading that CLPA does not reach.

N INJECTED AGAINST NATURAL MEMORIZATION

Much of the unlearning literature evaluates on secrets inserted by fine-tuning rather than on content the model memorized during pretraining. We asked whether the two are the same thing to a probe. A target string was injected into Pythia-70M by fine-tuning until its log-probability per token exceeded -0.3, and the natural-memorization pool was screened on the base model beforehand. A probe fit on the natural pool was then asked where the injected secret falls.

It does not transfer. Across all seven depths the natural-memorization probe scores the injected target at 0.500, except at depth one where it reaches 0.521. Behavioral erasure of the injected target succeeds over the same run, with its log-probability falling from -0.007 to -70.95, so the null is not an artifact of a failed injection.

Injecting five secrets at once and running leave-one-out across them does not help. The pure-distinguishability null reaches 0.683 while the memorized-class accuracy reaches 0.514, so the matched decoys, which differ only in their final tokens, are more reliably separable than anything the five injected secrets share. Rapid fine-tuning at this scale does not build cross-secret linear structure for a probe to find.

The reading is narrow and worth stating precisely. This is a null result for transfer and a positive result for the distinction. At 70M parameters, natural pretraining memorization and fine-tuning-injected memorization produce different linear-probe signatures in the same model, which is a reason to be careful about carrying conclusions from injected benchmarks over to pretrained content.

O RELEARNING AND EXTRACTION

Two attacks bear on whether behavioral suppression is shallow, and they answer differently. Both run on the controlled testbed, with recovery defined as $P(\text{secret} \mid \text{prefix}) > 0.90$.

Under direct relearning, gradient descent on the secret itself, the secret comes back almost immediately from every edit locus. Table 13 gives the step counts. The value edit used in this paper is the fastest to reverse of the four edit loci, at a single step against two for the unedited model.

Table 13: Steps to recover the secret by edit locus.

Edit locus	Steps to recovery
Unedited model	2
Value matrix	1
Feedforward block	3
Value and feedforward	3
Embedding	2

None of these edits confers relearning resistance, and a reader who expected behavioral unlearning to make a secret hard to restore should revise that expectation downward. The finding also sits with the thesis of the paper rather than against it. A secret that returns in one gradient step was not removed, and the probe agrees, since it reads the edited model at 1.000 throughout.

Indirect extraction is the other case. Fine-tuning on background domain text, with the secret never shown, drives the unedited model to $P = 0.99998$ at the first step, which is the model retaining what it already had. The edited model stays below 0.0026 across the fifty-step run and a randomly initialized control below 0.016, so neither reaches the recovery threshold. Behavioral suppression therefore survives incidental training that does not touch the secret and fails against training that does.

P STRONGER BEHAVIORAL BASELINES

Two methods deserve separate treatment because each is stronger than the five published objectives of Section 4.3 in a different way. Both run on the controlled testbed of Appendix G, with the same cross-sequence protocol and the same nine memorized prefixes.

Weight editing. MEMIT edits the parameters that store an association instead of optimizing against the model's output. On this testbed it drives recall four orders of magnitude below the base model at a perplexity cost of a third, and it leaves the probe at exactly its starting value. Table 14 also reports what it costs to buy probe suppression on top of it by applying inference PGA with k directions at every depth. The probe falls from 1.000 to 0.656 and held-out perplexity rises from 1.901 to 8.670. Adding directions does move the readout, and the price of moving it this way is a model with six times the perplexity of the unedited one.

Table 14: MEMIT with inference PGA, testbed.

Configuration	$\min P(\text{sec})$	PPL	Max probe
Base model	9.8×10^{-1}	1.403	1.000
MEMIT	8.4×10^{-5}	1.901	1.000
MEMIT + inference PGA, $k = 5$	6.4×10^{-4}	1.815	0.828
MEMIT + inference PGA, $k = 10$	4.7×10^{-7}	2.091	0.794
MEMIT + inference PGA, $k = 20$	9.8×10^{-8}	4.579	0.678
MEMIT + inference PGA, $k = 30$	1.7×10^{-9}	8.670	0.656

Distillation from a clean teacher. The teacher is a second network of the same architecture, initialized from a different seed and trained on all eighteen prefixes with a freshly sampled continuation at every epoch, so no fixed string is ever stored. It is the model the unlearning literature is implicitly trying to recover. The student is the memorized model, distilled toward the teacher by minimizing the Kullback-Leibler divergence to the teacher's next-token distribution on those same batches. The memorized continuations never appear in the distillation data.

Table 15 gives the per-depth readings. The student ends the procedure behaving like the teacher and reading like its former self. Depth three is the sharpest case: the teacher sits at 0.589 and the distilled student at 0.922, on a model whose recall has fallen by four orders of magnitude.

Table 15: Probe accuracy by depth after distillation.

Model	$\min P(\text{sec})$	PPL	D0	D1	D2	D3	D4
Memorized student	9.8×10^{-1}	1.403	0.656	0.739	0.883	1.000	1.000
Clean teacher	7.2×10^{-6}	1.404	0.661	0.656	0.633	0.589	0.533
Distilled student	2.6×10^{-5}	1.405	0.656	0.733	0.867	0.922	0.917

The reading is that output behavior does not carry the geometry that produces it. Two models can agree on what they emit and disagree about what a probe recovers from their residual streams, and imitation transfers the first without the second. For an auditor this is the same warning the rest of the paper gives, arriving by the route that looks safest.

Q RAW AND CORRECTED VERDICTS

Section 4.3 notes that alignment can invert a readout rather than remove what it reads. Table 16 audits ten configurations on the controlled testbed. It gives raw probe accuracy at each of the five depths, then the largest raw reading and the largest corrected reading, $\max(\text{acc}, 1 - \text{acc})$ (Burns et al., 2023), each taken over depths. The two maxima can come from different depths.

Table 16: Raw probe accuracy by depth and both maxima, testbed.

	Raw accuracy at depth					Largest reading	
Configuration	D0	D1	D2	D3	D4	Raw	Corrected
Base model	0.656	0.739	0.883	1.000	1.000	1.000	1.000
MEMIT	0.656	0.739	0.867	0.994	1.000	1.000	1.000
Distillation from a clean teacher	0.656	0.733	0.867	0.922	0.917	0.922	0.922
CLPA, $\lambda = 10.0$	0.656	0.778	0.833	0.828	0.833	0.833	0.833
Inference PGA, $k = 5$	0.606	0.528	0.483	0.739	0.644	0.739	0.739
Isotropic alignment	0.656	0.694	0.683	0.667	0.617	0.694	0.694
MEMIT + inference PGA, $k = 30$	0.383	0.517	0.511	0.656	0.556	0.656	0.656
PGA, $\lambda = 0.1$	0.650	0.550	0.294	0.222	**0.189**	0.650	0.811
PGA, $\lambda = 1.0$	0.650	0.550	0.339	0.256	**0.222**	0.650	0.778
PGA, $\lambda = 10.0$	0.650	0.556	0.361	0.256	**0.222**	0.650	0.778

Seven of the ten read identically under both conventions, because their largest reading lies above chance, where correction leaves it untouched, and none of their few below-chance readings exceeds it once corrected. The three trained PGA rows are the only ones that differ, and they differ by 0.128 to 0.161. No threshold has to be chosen for this to matter. Whatever bar an auditor sets, the verdict on the other seven is the same either way, and the verdict on trained PGA depends on which number is reported.

The PGA rows show how the two maxima separate. For PGA at $\lambda = 0.1$, the largest raw reading, 0.650, is at depth zero, and every deeper depth falls below chance. Correction maps each of those deeper readings to $1 - \text{acc}$, so the depth-four reading of 0.189, in bold, becomes 0.811, which is the corrected maximum. The raw value of 0.650 is therefore not corrected into 0.811. The strongest signal was at depth four throughout, and the probe there pointed the wrong way.

Table 17 shows that this is not a property of the testbed. Each row is the configuration named in Appendix I, read at the depth that run evaluates.

At every scale the raw reading falls below chance, and at three of the four it falls far enough that an auditor reporting it would record near-total erasure. The corrected column says the opposite in the first three. The two columns are complements by construction, which is the point: the same run supports either headline, and only the corrected one describes what a reader who fixes the sign recovers.

Two Pythia-70M readings appear in this paper and they differ. The raw value behind Table 2, quoted in Section 4.3, is 0.000, and Table 17 gives 0.107. Both describe PGA aligned at depths one through

Table 17: Raw and corrected readings after alignment, four scales.

Scale	Configuration	Raw	Corrected
Testbed	depth four, full weights, $\lambda = 0.1$	0.189	0.811
Pythia-70M	depth six, LoRA r=16, four-seed mean	0.107	0.893
GPT-2 Medium	depth 21, RK-PGA k=2	0.061	0.939
Mistral-7B	depths 24 and 28, four-bit, LoRA r=8	0.417	0.583

six. The first is the probe of the unified baselines pipeline, applied identically to all six rows, and the second is the leave-one-out protocol of Section 3.1 averaged over four adapter seeds. Each is the complement of its own corrected value, 1.000 and 0.893.

R SCOPE OF EACH CLAIM

Table 18 states the scale and configuration behind each claim.

Table 18: Scope of each claim.

Claim	Scale	Variant	Evidence
Cross-sequence signature present	70M–7.24B	n/a	Fig. 2
Verified-membership pool, $n = 30$	70M	n/a	Fig. 7
Signature size-stable	70M–1B	n/a	Fig. 6
Probe direction separable from recall	70M	n/a	Fig. 5
Probe moved at every scale	all four	all	Fig. 9
Best recall-per-capability trade-off	70M	PGA	Table 2
59.4 nats per nat against isotropic	70M	PGA	Table 3
Margin holds at matched penalty weight	70M	PGA	Table 3
Fresh linear pooled probe near the floor	70M	MP-PGA	Fig. 4
Five benchmarks within 2.9 points	70M	inference PGA	Table 8
Recall -2.63 nats at 3.6 nats/nat	7.24B	PGA	Section 4.6
No single sequence carries the gap	70M	n/a	Appendix K
Dissociation across eight secrets	testbed	n/a	Table 11
Head-local alignment insufficient	testbed	CLPA	Table 12
Injected memorization reads differently	70M	n/a	Appendix N
No relearning resistance	testbed	n/a	Table 13
Weight editing leaves the probe	testbed	MEMIT	Table 14
Distillation leaves the probe	testbed	distillation	Table 15
Only trained PGA's verdict depends on the metric	testbed	all	Table 16
Raw reading below chance at every scale	all four	all	Table 17