HIDDEN, NOT DELETED: HOW NETWORKS SUPPRESS ENTANGLED FEATURES

Anonymous authors
Paper under double-blind review

ABSTRACT

Concept erasure methods that operate via linear projection assume that features occupy separable subspaces. We show this assumption fails under dense superposition: when two features are forced into an antipodal pair sharing a single subspace, state-of-the-art linear erasure destroys both, not just the target. Networks trained with gradient descent instead solve this problem non-linearly, but not uniformly: they converge to one of two distinct circuit-level solutions depending on initialization, which we call mirror and shadow solutions. We map this bifurcation as a function of feature entanglement, show it reflects a stable attractor structure rather than an artifact of our setup, and use targeted causal interventions to demonstrate that both solutions leave a substantial, measurable trace of the erased feature's representation intact, recoverable through a single scalar patch rather than requiring any further training. This mirrors a failure mode recently observed empirically in LLM unlearning, where suppression rather than deletion allows forgotten knowledge to resurface; our results offer a mechanistic, causally-validated account of why that failure mode occurs.

1 INTRODUCTION

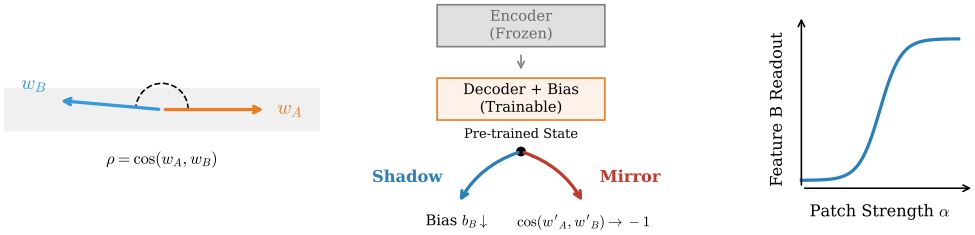
Machine unlearning methods, procedures intended to remove specific knowledge or capabilities from a trained model without full retraining (Cao & Yang, 2015; Bourtoule et al., 2021), are typically evaluated by whether a target behavior disappears. If a fine-tuned model no longer reproduces memorized text, or no longer exhibits a capability it was trained to lose, the method is judged successful. This standard is easy to apply and increasingly important in practice, since unlearning is now asked to do real work: removing memorized personal data, revoking specific capabilities, complying with deletion requests. Recent evidence suggests this standard may not be sufficient. Whether a behavior disappears is not the same question as whether the underlying knowledge was removed.

Yang et al. (2026) find that widely used unlearning methods in large language models tend to produce what they call spurious unlearning neurons, components that actively suppress a target output rather than removing the representation that produced it. Xu et al. (2025) report a complementary finding from a different angle: unlearned behavior in LLMs is frequently recoverable with a small amount of additional fine-tuning, which would not be possible if the underlying knowledge had genuinely been deleted. Read together, these results point at the same conclusion. A model can appear to have forgotten something while the information remains present, one nudge away from returning.

Both findings rest on behavioral evidence at scale. Neither isolates the underlying mechanism by targeting a specific internal quantity and demonstrating causal sufficiency for the apparent forgetting. Meeting this standard requires a fully tractable setting.

We study concept erasure in the Toy Model of Superposition (TMS) introduced by Elhage et al. (2022). TMS shows that when a network must represent more features than it has dimensions, and those features are sparse enough to rarely co-occur, the network learns to superpose several features onto the same direction, since they are rarely read out at once. A particularly clean instance of this is the antipodal pair, in which two features sit at exactly opposite points along a single shared axis. Erasing one antipodal feature with a linear method, the class of representation surgery behind tools such as LEACE (Belrose et al., 2023), fails immediately. We establish this failure formally and confirm it empirically in §3.

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.



(a) (b) (c)

Figure 1: Overview of the erasure-and-recovery pipeline studied in this paper. (a) An antipodal pair A, B sharing a single subspace, parameterized by entanglement $\rho = \cos(w_A, w_B)$. (b) Excision, with the encoder frozen and only the decoder and output bias trainable, bifurcates a single erasure objective into mirror and shadow solutions. (c) A single scalar causal patch (α, patch strength) restores the suppressed readout for feature B; the curve shown is the shadow solution's dose-response (§5.2), the mirror solution's recovery is a discrete absolute-value patch rather than a graded curve.

Training the network directly to suppress the unwanted feature, rather than projecting it away after the fact, succeeds where the linear approach fails, but not in one way. Depending on initialization, the network converges to one of two distinct solutions, and using a single targeted causal intervention rather than an inference from downstream behavior, we show that neither solution deletes the suppressed feature. Both hide it. We call this Janus erasure, after the two-faced god of thresholds, since a single erasure objective is resolved by one of two different internal mechanisms depending on where training begins.

This paper makes three contributions. First, we show analytically and empirically that linear erasure fails on entangled representations, with the failure worsening predictably as entanglement increases. Second, we characterize what gradient-based erasure does instead: it bifurcates into two attractors whose relative prevalence is a sharp function of entanglement, mapped precisely rather than observed qualitatively. Third, we show by direct causal intervention that both attractors leave the erased feature substantially recoverable through a single scalar patch, rather than genuinely deleted. If the same pattern holds in larger systems, and the empirical results above suggest it might, then judging unlearning by whether a behavior disappears may not be sufficient. Figure 1 previews the full pipeline these three contributions establish.

2 Related Work

Concept erasure. A substantial line of work removes a target concept from a representation without retraining the underlying model, motivated originally by fairness concerns such as gendered structure in word embeddings (Bolukbasi et al., 2016). The earliest approaches are linear: Iterative Nullspace Projection (Ravfogel et al., 2020) repeatedly projects onto the nullspace of a learned linear classifier, and LEACE (Belrose et al., 2023) gives a closed-form projection that is provably optimal among linear operators with respect to a whitened metric derived from the representation's covariance. Our linear baseline (§3.3) is a rank-1 nullspace projection in the same class as LEACE's action for a single direction, though without its covariance-whitening step; §3.3 states the precise relationship between the two. Linear methods are known to be vulnerable to nonlinear adversaries, which motivated R-LACE (Ravfogel et al., 2022a) and later nonlinear alternatives such as kernelized concept erasure (Ravfogel et al., 2022b) and kernelized rate-distortion maximization (Basu Roy Chowdhury et al., 2023). These methods ask whether a post-hoc transformation can separate a concept after training. We ask a related but different question: what mechanism gradient-based training itself discovers when tasked with suppressing a feature, since training-time optimization, not post-hoc surgery, is what most practical unlearning pipelines rely on. We return to this distinction, and to why a nonlinear method could in principle succeed here where linear projection cannot, in §7.

Suppression versus deletion in LLM unlearning. Our results echo a failure mode recently documented empirically in large language model unlearning. Yang et al. (2026) show that widely used unlearning methods induce shallow alignment: rather than erasing target knowledge, they produce spurious unlearning neurons that suppress it, leaving it recoverable under further fine-tuning. Xu

et al. (2025) reach a similar conclusion from a different direction, showing that unlearned behavior in LLMs is frequently reversible with minimal additional training. Both results are established empirically, at LLM scale, without a controlled account of the underlying mechanism. Our contribution is a candidate mechanistic account: a minimal setting where the hide-versus-delete distinction is established exactly, and recoverability is shown by direct causal intervention rather than inferred from downstream relearning.

Superposition and polysemanticity. Beyond the original TMS characterization (Elhage et al., 2022), subsequent work has used the same superposition framing to motivate sparse dictionary learning for decomposing polysemantic activations in real language models into interpretable, monosemantic features (Bricken et al., 2023; Cunningham et al., 2023). Our use of TMS is oriented differently: rather than decomposing superposed representations for interpretation, we study what happens when a specific pair of already-superposed features becomes the target of an erasure objective.

Causal intervention as evidence. Establishing that a model's behavior is caused by a specific internal quantity is standard practice in mechanistic interpretability, building on circuit-style analysis of trained networks (Olah et al., 2020; Elhage et al., 2021; Olsson et al., 2022). Causal mediation analysis (Vig et al., 2020) and causal abstraction (Geiger et al., 2021) formalize this style of intervention, and causal tracing (Meng et al., 2022) showed that intervening on individual hidden states is needed to identify which components of a network are responsible for a given behavior. Subsequent work has used activation patching to localize circuits underlying specific behaviors in language models (Wang et al., 2022) and to automate the search for such circuits (Conmy et al., 2023). Our causal patching methodology (§5) follows the same logic, applied to concept erasure rather than factual recall or circuit discovery, extended with explicit sufficiency checks and negative controls to verify our interventions are both necessary and minimal.

3 Setup: Antipodal Superposition and the Limits of Linear Erasure

To study the hide-versus-delete distinction in a setting precise enough for exact causal analysis, we build on the Toy Model of Superposition (TMS) of Elhage et al. (2022), and use it first to examine whether the most natural approach to erasure, linear representation surgery, is viable here. It is not, and this motivates the gradient-based approach that is our main subject, taken up in §4.

3.1 Toy Model of Superposition

A model with n input features and a bottleneck of dimension $d \ll n$ consists of an encoder $W_E \in \mathbb{R}^{d\times n}$ and a decoder $W_U \in \mathbb{R}^{n\times d}$ with output bias $b_{\text{out}} \in \mathbb{R}^n$:

$$h = W_E x, \qquad \hat{x} = \mathrm{ReLU}(W_U h + b_{\text{out}}), \tag{1}$$

where $\mathrm{ReLU}(z) = \max(0, z)$ (Nair & Hinton, 2010), trained to minimize reconstruction error $\mathbb{E}_x\|\hat{x} - x\|_2^2$ on sparse input vectors x, where each coordinate is independently zero with probability S and otherwise drawn from a fixed magnitude distribution. Elhage et al. (2022) show that under sparsity, a network represents more features than it has bottleneck dimensions by packing several features into overlapping directions in h-space, a phenomenon they call superposition. When two features never co-occur, the cheapest solution places them at antipodal points on a shared one-dimensional subspace, $\cos(w_A, w_B) = -1$, where w_A and w_B are the corresponding columns of W_E. Throughout this paper, entanglement is designated and controlled for exactly one pair of features, A and B; the remaining $n - 2$ features are initialized and trained without any geometric intervention, so their pairwise relationships (including relative to A and B) are whatever ordinary training under sparsity produces, not additionally engineered. We confirm in §4.4 that antipodal geometry for a designated pair arises under realistic training conditions rather than being a case we construct. The independent sampling described above applies to the $n - 2$ background features; for the designated pair A, B, co-occurrence is instead controlled directly through an anti-correlation parameter (the same `anti_corr_prob` used in the natural-emergence sweep of §4.4), which at its maximum enforces strict mutual exclusivity, $x_A \cdot x_B = 0$ for every input, and which is what makes the antipodal analysis in the remainder of this section exact rather than approximate.

3.2 ENTANGLEMENT AND THE ERASURE TASK

For a designated pair of features A and B, we call the cosine similarity between their encoder directions, $\rho = \cos(w_A, w_B) \in [-1, 1]$, their *entanglement*, and treat it as the main variable throughout this paper. Given a model with entanglement ρ, we ask whether B's output can be suppressed for every input while A's reconstruction is left exactly as it was. We track two quantities: erasure, the mean output magnitude for B, which should fall to zero, and preservation, A's absolute reconstruction MSE after the intervention, $\mathbb{E}_x[(\hat{x}_A - x_A)^2]$, which should stay close to its pre-intervention value. Because the model represents A in superposition with $n - 2$ other features besides B, this quantity is not zero even before any intervention; the $\rho = 0$ value in Table 1 (0.098) is this ordinary interference floor, not damage caused by erasure. What indicates a preservation failure is a *rise* above that floor as entanglement increases, not the floor's absolute value. A method that improves erasure at the cost of raising preservation MSE above this floor has not solved the task.

3.3 LINEAR ERASURE FAILS

The most direct way to erase a concept from a representation is rank-1 nullspace projection, closely related to LEACE's action for a single target direction (Belrose et al., 2023) (see §2 for the precise relationship between the two). It removes B by projecting the bottleneck activation onto the orthogonal complement of B's own direction:

$$h' = h - (h \cdot \hat{u})\,\hat{u}, \qquad \hat{u} = w_B / \|w_B\|. \tag{2}$$

This succeeds when A and B occupy different directions, but not in the antipodal case we study. When $\rho = -1$, w_A and $\hat{u}$ are collinear, so the projection removes any component of h along w_A together with w_B. There is no direction that carries information about B without also carrying A, because up to sign they are the same direction. Table 1 confirms this holds in practice across the full entanglement range, not only at $\rho = -1$: erasure of B succeeds throughout, while preservation of A degrades monotonically as entanglement increases, reaching $4.2\times$ the orthogonal-case error at full antipodal entanglement.

Table 1: Linear erasure destroys A's reconstruction as entanglement increases, while succeeding at erasing B. Mean over 5 seeds.

Entanglement ρ	**0.00**	-0.50	-1.00
Preservation MSE (A)	0.098	0.144	0.411
Residual activation (B)	~ 0	~ 0	~ 0

This geometric obstruction applies to any linear operator at $\rho = -1$, regardless of covariance whitening. At intermediate ρ (Table 1), the degradation is specific to our unwhitened rank-1 projection; optimally whitened linear operators might yield quantitatively different preservation curves, which we leave to future work. However, these values confirm the failure is not a knife-edge artifact of exact antipodality. We focus instead on what gradient-based training discovers when tasked with erasing B, and whether it genuinely deletes or merely suppresses the feature.

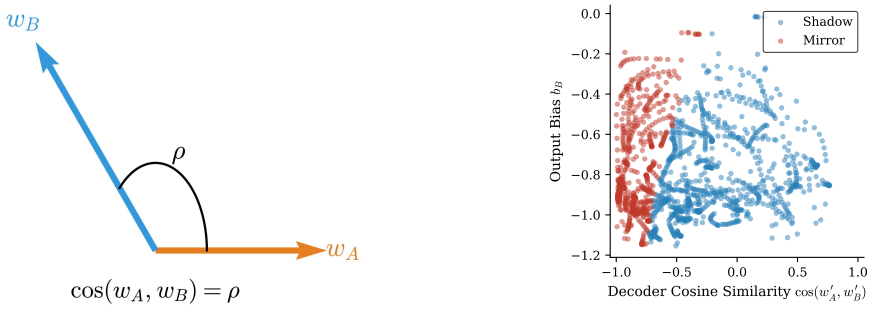
4 THE BIFURCATION PHENOMENON

As shown in §3.3, linear projection fails because there is no direction left to remove one feature without the other. Training is not restricted in this way: it has access to a nonlinearity, and it is not required to find a single fixed transformation that works for every input at once. We examine what happens when the network is optimized directly to suppress B.

4.1 EXCISION PROCEDURE: FREEZING THE ENCODER

Starting from a model with encoder geometry fixed at entanglement ρ, we fine-tune the decoder and output bias to minimize

$$\mathcal{L} = \mathbb{E}_x \sum_i (\hat{x}_i - x_i)^2 + \lambda \mathbb{E}_x\, \hat{x}_B^2, \tag{3}$$



(a) (b)

Figure 2: (a) The antipodal geometry under study, parameterized by entanglement $\rho = \cos(w_A, w_B)$. (b) Basin of attraction over all seeds and entanglement conditions, colored by classified basin.

holding the encoder W_E fixed throughout. The reconstruction term is evaluated over all features, including B. In our implementation, the reconstruction objective utilizes a mean reduction over the 16-dimensional feature space, resulting in an effective excision penalty weight of $\lambda = 16$ relative to the unreduced sum. This concrete weighting forces the network to prioritize the suppression of B, driving the target output to a theoretical minimum of $x_B/17$, which aligns with the near-zero residual activations observed empirically. An unfrozen encoder gives the network a second way to satisfy the excision term: it can simply stop encoding B at all, which trivially minimizes the loss without revealing how an existing representation gets suppressed at readout. We observed exactly this failure mode in an early version of the experiment (Appendix E). Freezing W_E isolates the question of interest: given that B is faithfully represented, how does the network suppress it?

4.2 Two attractors: mirror and shadow

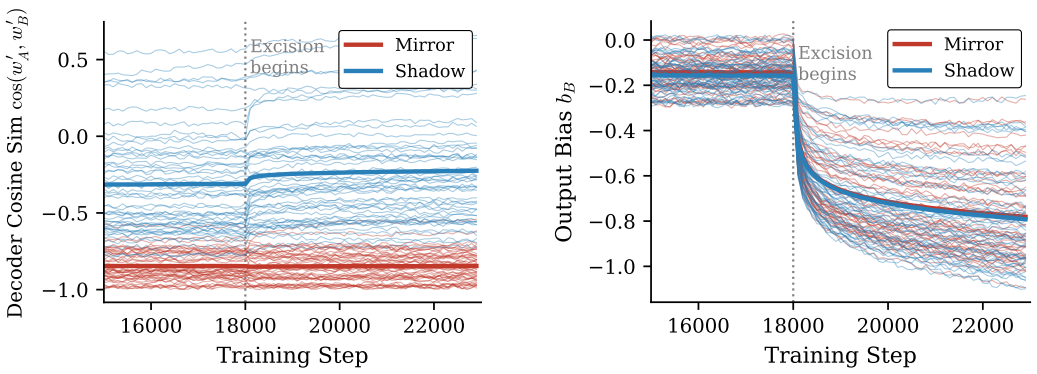
We sweep $\rho \in [-1, 0]$ at 15 points with 100 seeds each (grid construction and compute budget in Appendix B), and for every run record the final decoder cosine similarity $\cos(w'_A, w'_B)$ and the final output bias b_B, where w'_A, w'_B denote the decoder rows at excision-phase convergence, distinct from the fixed encoder columns w_A, w_B (formal definition in Appendix A.1). Figure 2 plots every run's endpoint, colored by a Gaussian mixture fit to this joint distribution (Appendix B). Two clusters separate clearly. In the mirror solution, the decoder direction for B rotates substantially, approaching full reflection. In the shadow solution, the decoder direction rotates only partially, and the output bias is driven strongly negative instead. We use "basin" and "attractor" informally throughout this paper, to denote the two stable solution types training consistently converges to across seeds, not in the sense of a formally proven dynamical-systems attractor with an associated stability proof.

Mirror-basin runs reach a median final cosine similarity of -0.86, close to full reflection. Shadow-basin runs range from near zero at low entanglement to a mean of -0.40 at $\rho = -1$ (Appendix C.1 gives the exact per-basin means). At $\rho = -1$, the decoder cosine similarity immediately before excision begins is -0.80 on average, itself already substantially antipodal; a strictly bias-only account, with the decoder direction left untouched, would predict both basins remain near this pre-excision value. Neither does: mirror moves further negative, deepening the pre-existing antipodal alignment toward full reflection, while shadow moves substantially less negative, undoing much of it. Bias suppression is substantial in both basins, not only in shadow. We read this as one continuous mechanism, decoder rotation combined with bias suppression, in which mirror uses more of the rotation and shadow uses less.

The basin scatter in Figure 2 shows only endpoints. Figure 3 shows the bifurcation as it happens during excision, rather than only its converged outcome.

4.3 A phase transition in basin prevalence

The population-level outcome is nonetheless sharp as a function of entanglement (Figure 4). At full antipodal entanglement, 75% of runs land in mirror; by $\rho = 0$, this reverses to 87% in shadow, with the crossover at $\rho \approx -0.75$. A broadly similar pattern, in which a sharp population-level



(a) (b)

Figure 3: Because $\rho = -0.75$ is near the crossover, trajectories overlap substantially; basin labels are assigned via the Gaussian mixture classifier at convergence (§4.2, Appendix B), not inferred visually. Visible separation before the excision marker suggests the outcome is sensitive to pre-excision initialization rather than determined solely by post-excision dynamics; we do not intend the marker to imply identical starting states.

transition sits on top of training dynamics that look arbitrary at the level of any single run, has been reported in the study of grokking (Power et al., 2022; Nanda et al., 2023; Barak et al., 2022), though the underlying dynamics differ. Basin fractions are reported with Wilson 95% confidence intervals (Wilson, 1927) at each of the 15 swept values. Unlike the linear baseline, whose preservation cost for A rises sharply with entanglement (Table 1), gradient-based excision holds A's preservation MSE essentially flat across the same sweep, in both basins (Table 2): the bifurcation in how B is suppressed does not come at the cost of A's reconstruction.

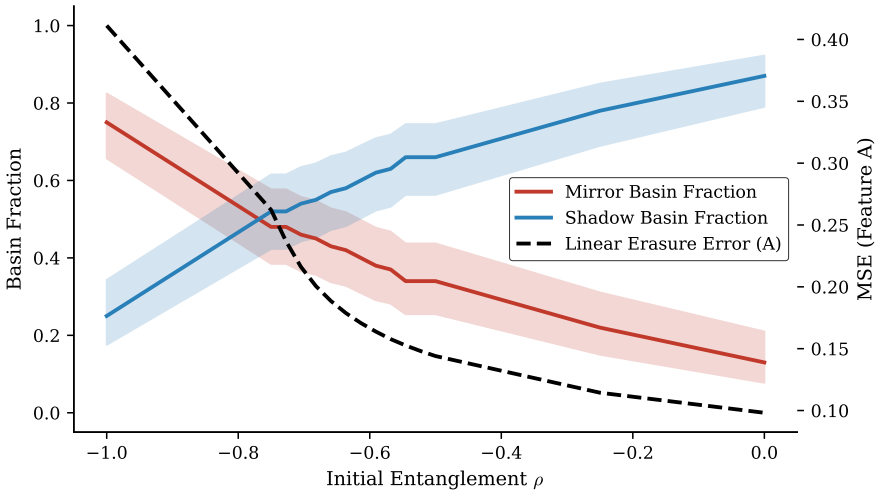


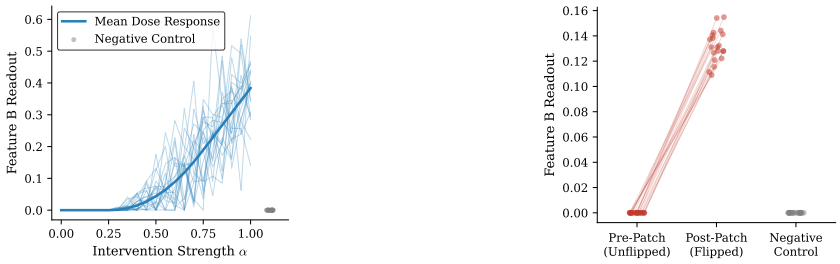
Figure 4: Basin fraction as a function of entanglement, with Wilson 95% confidence intervals. Linear baseline preservation MSE for A (Table 1) overlaid for reference: linear erasure's cost to A rises as entanglement increases, exactly where nonlinear erasure's basin structure is most sharply bifurcated. Gradient-based excision's own preservation MSE for A across this same sweep, by contrast, stays essentially flat and close to the interference floor in both basins (Table 2), the direct comparison behind the claim that nonlinear excision does not trade preservation for erasure the way linear projection does.

4.4 Natural emergence of antipodal geometry

While entanglement is fixed by hand throughout this section to isolate it as a variable, we separately train models with no geometric intervention to confirm that entanglement this extreme arises under

Table 2: Gradient-based excision preserves A's reconstruction far more tightly than linear erasure across the full entanglement sweep, in both basins. Compare to Table 1, where linear erasure's preservation MSE for A rises from 0.098 to 0.411 over the same range.

Entanglement ρ	**0.00**	-0.25	-0.50	-0.75	-1.00
Preservation MSE (A), mirror	0.0190	0.0184	0.0190	0.0202	0.0207
Preservation MSE (A), shadow	0.0197	0.0201	0.0206	0.0207	0.0198



(a) (b)

Figure 5: Causal patching recovers B's output. (a) Shadow solution, bias dose-response: individual seeds (thin lines) and mean response (bold) as patch strength α increases. (b) Mirror solution, absolute-value patch ($z_B \mapsto |z_B|$): per-trial readout before and after the patch. In both panels, negative controls (patching a randomly chosen, unrelated feature, 20 trials per basin) are shown as individual trial outcomes and produce zero measurable effect on B's readout (§5.3).

ordinary training. We sweep sparsity and anti-correlation strength, and track how often antipodal geometry ($\rho < -0.9$) emerges on its own. It emerges frequently, and in the pattern Elhage et al. (2022) describe: more often as anti-correlation strengthens and as the bottleneck narrows relative to the number of features (full sweep in Appendix F, Figure 7).

5 Causal Validation

Suppressing B's output to near zero shows that B is no longer expressed. It does not show that B has been removed. We test the difference directly, with a single targeted intervention.

5.1 Patching procedure: a single targeted intervention

For a representative high-confidence seed from each basin (posterior probability > 0.9 under the basin classifier), we apply one targeted scalar intervention and measure its effect on B's readout, holding every other parameter fixed. For the shadow solution, we sweep the output bias b_B from its trained, suppressed value back toward its pre-excision value, $b_B(\alpha) = (1-\alpha)\, b_B^{\text{trained}} + \alpha\, b_B^{\text{pre-excision}}$, $\alpha \in [0, 1]$. For the mirror solution, we replace B's pre-activation with its absolute value, $z_B \mapsto |z_B|$, which restores a positive pre-activation whenever excision had driven it negative. Both interventions are pure forward-pass overrides; neither mutates a stored parameter.

5.2 Erasure is reversible

In the shadow solution, B's mean readout rises smoothly from 0 to 0.38 as $\alpha \to 1$ (Figure 5), a graded recovery rather than a discontinuous jump. The mirror solution's absolute-value patch recovers B's readout from 0 to 0.13 in one step. In neither case was any parameter beyond the single targeted scalar touched.

5.3 Sufficiency and specificity

Two checks rule out an incidental or diffuse effect. Sufficiency: patching the single targeted scalar fully accounts for the recovery, with zero measurable change to any other feature's readout. Due

to the additive, per-feature bias architecture, Δb_B is strictly independent of $b_{i \neq B}$, making this a structural property of the model rather than an empirical coincidence. Specificity: applying the identical intervention to a randomly chosen, unrelated feature, 20 trials per basin, produces zero effect on B's readout (Figure 5, negative control).

In both attractors, the suppressed feature remains detectable through a single interpretable intervention: the patch produces a substantial, non-zero readout for B using only the targeted scalar, indicating the representation was suppressed at readout rather than removed from the network.

6 Robustness Across Architectures and Nonlinearities

The results above come from a minimal two-layer, ReLU network with a single antipodal pair, isolated as described in §3.1 so that entanglement between A and B is the only geometric quantity under our control. We check whether the phenomenon depends on these specific choices. Initial experiments on a small-scale causal transformer similarly show plausible generalization beyond the autoencoder setting (Appendix G).

6.1 Depth: an additional hidden layer

We repeat the excision procedure with an additional hidden layer between the bottleneck and the readout, keeping the encoder frozen as before. The bifurcation into mirror and shadow persists. Suppression is concentrated mainly in the output layer: across the swept entanglement range, 91.0% of seeds show output-layer-dominant suppression and the remaining 9.0% show hidden-layer-dominant suppression, with no seeds showing a split between the two (full breakdown in Appendix D, Table 6).

6.2 Nonlinearity: replacing ReLU with GELU

We repeat the sweep with GELU (Hendrycks & Gimpel, 2016) in place of ReLU. GELU has no hard zero region, so we define erasure functionally: a feature counts as erased if a causal patch restoring its pre-excision parameters produces an effect exceeding what an identical patch produces on a matched control feature (Appendix B). Suppression still occurs reliably under this definition. The basin structure is less clean than under ReLU: a Bayesian information criterion (Schwarz, 1978) comparison favors four mixture components rather than two, and the output bias stays close to zero across the entanglement sweep instead of moving with entanglement the way it does under ReLU. We take this as GELU's soft negative region changing which suppression strategies are available, rather than a failure of the underlying method.

Figure 6 summarizes both robustness checks together against the ReLU depth-1 baseline of §4.2–§4.3.

7 Conclusion and Discussion

We studied concept erasure in a setting small enough to trace exactly: two features sharing a single antipodal subspace in a toy model of superposition. While linear erasure fails here, gradient-based erasure succeeds, but converges to one of two attractors, a near-complete geometric reflection or a partial reflection paired with output suppression, and which one dominates is governed by entanglement in a way that is sharp at the population level even though individual runs look arbitrary. In both attractors, a single causal intervention shows that the suppressed feature remains recoverable rather than deleted.

Relation to nonlinear post-hoc erasure. A nonlinear erasure method could plausibly separate A and B where linear projection cannot. Because A and B never co-occur and sit at opposite points on the same axis, the sign of the activation along that axis still identifies which feature is present, even though a linear projection removes this information along with everything else. This is close to what the mirror solution does on its own, using the ReLU nonlinearity to build a sign-based gate through ordinary gradient descent. Methods such as kernelized concept erasure (Ravfogel et al., 2022b) and kernelized rate-distortion maximization (Basu Roy Chowdhury et al., 2023) address a related but distinct question: whether a post-hoc transformation, applied after training, can separate a concept

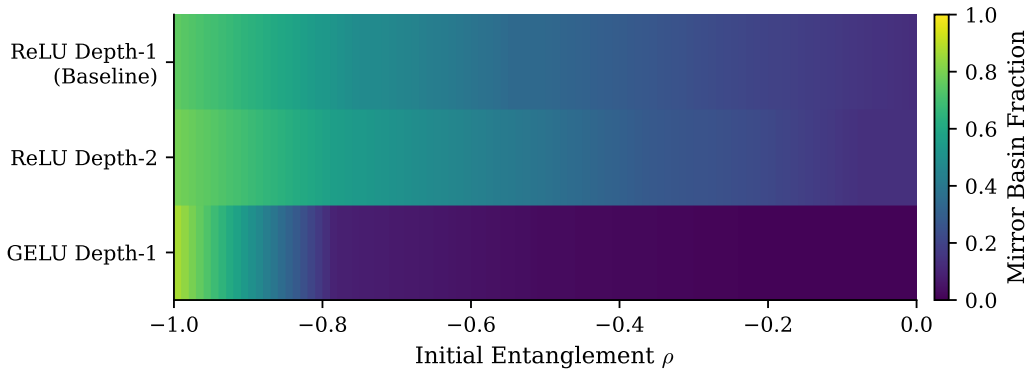


Figure 6: Mirror-basin fraction as a function of entanglement, across architecture variants. The crossover persists under an additional hidden layer (§6.1) and under GELU (§6.2), though the transition is sharper and confined to a narrower range of ρ under GELU, consistent with the less-clean two-basin structure reported there.

from a fixed representation. We instead characterize what training-time optimization discovers on its own, and whether that discovery constitutes genuine erasure. The two questions are complementary, and our results speak only to the second.

Relation to unlearning in practice. The pattern we describe provides a controlled candidate account for empirical unlearning failures at the scale of real models, where methods that appear to remove target knowledge can leave it recoverable (Yang et al., 2026; Xu et al., 2025). We provide a candidate account in a setting where it can be checked directly, by intervention rather than by relearning after the fact. If a similar mechanism operates in larger models, evaluating unlearning by whether a behavior disappears may not be a sufficient standard on its own.

Limitations. Our results come from a toy autoencoder with a single antipodal pair, chosen because it is small enough to characterize exactly, not because it resembles a production model. The transformer check in Appendix G suggests the phenomenon is not restricted to the autoencoder setting, but is not evidence that the same mechanism operates in language models at scale. We also study one antipodal pair in isolation, while production models must suppress many overlapping concepts at once, and it is not yet known whether the mirror and shadow attractors remain distinct once several such pairs share a network. Three untested comparisons would sharpen the practical relevance of this account: whether post-hoc nonlinear erasure such as kernelized concept erasure (Ravfogel et al., 2022b) or kernelized rate-distortion maximization (Basu Roy Chowdhury et al., 2023) avoids the bifurcation entirely, whether it persists under unlearning objectives drawn from the literature, such as representation misdirection (Li et al., 2024), rather than our bespoke excision loss, and whether anything resembling it is visible on a standard benchmark such as TOFU (Maini et al., 2024) or Bias in Bios (De-Arteaga et al., 2019) rather than hand-constructed antipodal data. We see these as the most direct next steps for testing whether this account has practical teeth beyond the setting we constructed it in; Appendix I discusses each in more detail.

A further limitation follows directly from our own methodology: we freeze the encoder W_E specifically because, when it is left trainable, gradient descent finds a third route that neither mirror nor shadow uses, shrinking B's encoder column toward zero (Appendix E), which comes far closer to genuine deletion than to suppression. Most practical unlearning setups, including full fine-tuning of a language model, do not hold earlier layers fixed the way we do here, raising a question our results do not answer: why would suppression neurons appear in practice (Yang et al., 2026) if genuine deletion is available whenever upstream parameters are free to move? We do not know, but Appendix E discusses two hypotheses for why deletion may be less available in practice than this comparison implies.

AI USE STATEMENT

In this work, we utilized generative AI tools strictly as an assistant for manuscript preparation and literature discovery. We used AI to help structure outlines, draft localized portions of the text, and polish our prose across revision passes, with the authors maintaining full editorial control over the final wording. Additionally, AI assisted in retrieving related literature discussed in §2, and provided technical support with LaTeX typesetting, including generating the structural formatting for our plots and figures.

Crucially, AI assistance was limited to writing and formatting support. It was not used for any core research ideation, mathematical derivations, experimental execution, or synthetic data generation. All theoretical framing, methodological design, and interpretation of results were independently directed and executed by the human authors. We take full responsibility for the final content of this manuscript, having thoroughly verified all code, raw experimental outputs, and primary citations prior to submission.

ETHICS STATEMENT

This work uses only synthetic data generated by controlled experimental procedures described in this paper; it does not involve human subjects, personally identifiable information, or the release of any dataset derived from real individuals. We are not aware of a direct conflict of interest or sponsorship relevant to this submission.

We note one dual-use consideration relevant to the paper's subject matter. This work characterizes a mechanism by which gradient-based concept erasure can leave suppressed information latent and recoverable rather than genuinely deleted. We view this as primarily defensive in orientation, since it identifies a failure mode in current erasure and unlearning practice that a defender would want to know about, consistent with the empirical findings it builds on (Yang et al., 2026; Xu et al., 2025), both of which report the same failure mode in production systems without our work's involvement. The causal recovery technique used in this paper (§5) requires white-box access to model internals; it does not constitute a new attack against black-box deployed systems, and we are not aware of a plausible misuse pathway that this paper's specific toy-model results would meaningfully enable beyond what is already established in the cited empirical literature.

REPRODUCIBILITY STATEMENT

We describe the model, task, and excision procedure precisely in §3.1–§4.1 and Appendix A.1. Full experimental methodology, including grid construction, the compute-budget safeguard, the basin classification procedure, confidence interval computation, and the functional erasure definition used under GELU, is given in Appendix B. Complete numerical results supporting every figure and summary statistic in the main text are provided in Appendices C–G, including the full 15-point sweep underlying Figure 4 and Table 1. Of the two theoretical results, the linear-impossibility claim informally stated in §3.3 is proved in full in Appendix A.2, and the sign-separability claim informally stated in §7 is proved in full in Appendix A.3; the latter includes an explicit statement of the one non-trivial assumption used to connect the bottleneck-level result to output-level reconstruction error. The methodological correction described in Appendix E is reported in sufficient detail to be independently checked by comparing frozen- and unfrozen-encoder training. All code and experimental scripts used in this work will be made publicly available upon acceptance.

ACKNOWLEDGMENTS

This work draws on the interpretability framework and superposition analysis developed by Elhage et al. (2022) and Elhage et al. (2021). Models were implemented in PyTorch (Paszke et al., 2019); data processing and analysis used NumPy (Harris et al., 2020) and pandas (McKinney, 2010); the Gaussian mixture classifier and logistic regression probes used scikit-learn (Pedregosa et al., 2011); statistical tests used SciPy (Virtanen et al., 2020); figures were made with Matplotlib (Hunter, 2007).

REFERENCES

Boaz Barak, Benjamin L. Edelman, Surbhi Goel, Sham Kakade, Eran Malach, and Cyril Zhang. Hidden progress in deep learning: Sgd learns parities near the computational limit. *arXiv preprint arXiv:2207.08799*, 2022.

Somnath Basu Roy Chowdhury, Nicholas Monath, Kumar Avinava Dubey, Amr Ahmed, and Snigdha Chaturvedi. Robust concept erasure via kernelized rate-distortion maximization. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, 2023.

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)*, volume 36, 2023.

Tolga Bolukbasi, Kai-Wei Chang, James Y. Zou, Venkatesh Saligrama, and Adam T. Kalai. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 29, 2016.

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)*, pp. 141–159. IEEE, 2021.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nicholas Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E. Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023. URL https://transformer-circuits.pub/2023/monosemantic-features/index.html.

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.

Arthur Conmy, Augustine Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. Towards automated circuit discovery for mechanistic interpretability. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, 2023.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. *arXiv preprint arXiv:2309.08600*, 2023.

Maria De-Arteaga, Alexey Romanov, Hanna Wallach, Jennifer Chayes, Christian Borgs, Alexandra Chouldechova, Sahin Geyik, Krishnaram Kenthapadi, and Adam Tauman Kalai. Bias in bios: A case study of semantic representation bias in a high-stakes setting. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pp. 120–128, 2019.

Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Christopher Olah. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021. URL https://transformer-circuits.pub/2021/framework/index.html.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. Toy models of superposition. *Transformer Circuits Thread*, 2022. URL https://transformer-circuits.pub/2022/toy_model/index.html.

Atticus Geiger, Hanson Lu, Thomas Icard, and Christopher Potts. Causal abstractions of neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, 2021.

Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, 2020.

Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (GELUs). *arXiv preprint arXiv:1606.08415*, 2016.

John D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9 (3):90–95, 2007.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

Nathaniel Li, Alexander Pan, Anjali Gopal, Summer Yue, Daniel Berrios, Alice Gatti, Justin D. Li, Ann-Kathrin Dombrowski, Shashwat Goel, Long Phan, et al. The WMDP benchmark: Measuring and reducing malicious use with unlearning. *arXiv preprint arXiv:2403.03218*, 2024.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary C. Lipton, and J. Zico Kolter. TOFU: A task of fictitious unlearning for LLMs. *arXiv preprint arXiv:2401.06121*, 2024.

Wes McKinney. Data structures for statistical computing in Python. In *Proceedings of the 9th Python in Science Conference*, pp. 56–61, 2010.

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, 2022.

Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted Boltzmann machines. *Proceedings of the 27th International Conference on Machine Learning (ICML)*, 2010.

Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability. In *International Conference on Learning Representations (ICLR)*, 2023.

Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. Zoom in: An introduction to circuits. *Distill*, 2020. doi: 10.23915/distill.00024.001.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Christopher Olah. In-context learning and induction heads. *Transformer Circuits Thread*, 2022. URL `https://transformer-circuits.pub/2022/in-context-learning-and-induction-heads/index.html`.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12: 2825–2830, 2011.

Alethea Power, Yuri Burda, Harri Edwards, Igor Babuschkin, and Vedant Misra. Grokking: Generalization beyond overfitting on small algorithmic datasets. *arXiv preprint arXiv:2201.02177*, 2022.

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*, 2020.

Shauli Ravfogel, Michael Twiton, Yoav Goldberg, and Ryan D. Cotterell. Linear adversarial concept erasure. In *International Conference on Machine Learning (ICML)*, pp. 18400–18421. PMLR, 2022a.

Shauli Ravfogel, Francisco Vargas, Yoav Goldberg, and Ryan Cotterell. Adversarial concept erasure in kernel space. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6034–6055, 2022b.

Gideon Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464, 1978.

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.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. Investigating gender bias in language models using causal mediation analysis. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, 2020.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, and Paul van Mulbregt. SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nature Methods*, 17:261–272, 2020.

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. *arXiv preprint arXiv:2211.00593*, 2022.

Frank Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83, 1945.

Edwin B. Wilson. Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association*, 22(158):209–212, 1927.

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.

Nakyeong Yang, Dong-Kyum Kim, Jea Kwon, Minsung Kim, Kyomin Jung, and Meeyoung Cha. Erase or hide? suppressing spurious unlearning neurons for robust unlearning. In *International Conference on Learning Representations (ICLR)*, 2026. arXiv:2509.22263.

APPENDIX

The appendix follows the order of the main text. Appendix A fixes the notation used throughout and works out two derivations the main text only states informally. Appendix B specifies the experimental procedures, grid construction, the basin classifier, and related methodology, that every later appendix and several sections of the main text depend on. Appendices C through G give the full supporting evidence behind the paper's empirical claims, in the order those claims appear in the main text: the bifurcation phenomenon, robustness across architectures, the encoder-freezing correction that the robustness results rely on, natural emergence, and the transformer scale check. Appendix H closes with an exploratory account of why entanglement governs the bifurcation, offered as a direction for future work rather than a result of this paper.

A FORMAL SETUP AND THEORETICAL ANALYSIS

This appendix has two purposes. The first is to restate the model, the excision procedure, and the causal patch operators with enough precision that the derivations which follow, and the discussion in §7, can refer to them exactly rather than informally. The second is to work out two arguments the main text only sketches: how much damage linear erasure does to A as a function of entanglement, and why a nonlinear rule can succeed at the erasure task where no linear rule can.

A.1 FORMAL SETUP

Model. The encoder $W_E \in \mathbb{R}^{d\times n}$, decoder $W_U \in \mathbb{R}^{n\times d}$, and output bias $b_{\text{out}} \in \mathbb{R}^n$ define

$$h = W_E x, \qquad \hat{x} = \mathrm{ReLU}(W_U h + b_{\text{out}}). \tag{4}$$

For the designated pair (A, B), we write $w_A, w_B \in \mathbb{R}^d$ for the corresponding columns of W_E, and $u_A, u_B \in \mathbb{R}^d$ for the corresponding rows of W_U *prior to excision*, i.e. whatever ordinary pre-training produces before any excision fine-tuning begins. This is distinct from w'_A, w'_B, defined below, which denotes the decoder rows *after* excision-phase convergence; the prime marks the only quantities that excision fine-tuning is free to move, since the encoder is frozen throughout (§4.1). Entanglement is $\rho = \cos(w_A, w_B) = \frac{w_A \cdot w_B}{\|w_A\|\|w_B\|} \in [-1, 1]$; this is the quantity the two derivations below are stated in terms of.

Excision phase. Given a model with W_E fixed at some entanglement ρ, the excision phase optimizes only W_U and b_{out}, using AdamW (Loshchilov & Hutter, 2019), a decoupled-weight-decay variant of Adam (Kingma & Ba, 2015), against

$$\mathcal{L}(W_U, b_{\text{out}}) = \mathbb{E}_x \sum_i (\hat{x}_i - x_i)^2 + \lambda \mathbb{E}_x\, \hat{x}_B^2, \qquad W_E \text{ fixed}, \tag{5}$$

with an effective scaling factor of $\lambda = 16$ derived from the mean-reduced feature dimension. We write w'_A, w'_B for the rows of W_U at excision-phase convergence, to distinguish the post-training decoder directions from the fixed encoder columns w_A, w_B, and b_B for the corresponding trained output bias. Appendix E documents what happens to this procedure if W_E is left free instead.

Causal patch operators. Both interventions in §5 are defined as functions of the forward pass, not as parameter edits, so that neither ever mutates a stored weight. For the shadow solution, the bias patch is

$$\mathrm{patch}_{\text{bias}}(x, \alpha) = \mathrm{ReLU}\Big(W_U h + b_{\text{out}} + \alpha\big(b_B^{\text{pre-excision}} - b_B\big)\, e_B\Big), \qquad \alpha \in [0, 1], \tag{6}$$

where $h = W_E x$ and e_B is the standard basis vector for feature B; this reduces to the ordinary forward pass at $\alpha = 0$. For the mirror solution, the absolute-value patch replaces the B-th pre-activation with its absolute value before applying the nonlinearity, leaving every other coordinate of the pre-activation vector untouched. This coincides with a literal sign flip whenever the pre-patch pre-activation is negative, which is the case throughout the mirror basin by construction; we did not separately audit its effect on inputs where A is active and B's pre-patch pre-activation happens to be positive, since the sufficiency and specificity checks of §5.3 evaluate the patch's effect on other features' readouts rather than on this specific edge case.

Basin label. A run's basin label is assigned after training, from the fitted point (w'_A, w'_B, b_B) at excision-phase convergence, by the classifier described in Appendix B.3, not from any property of training visible during the run itself. (We use "basin" and "attractor" informally throughout this paper, to denote the two stable solution types training consistently converges to across seeds, not in the sense of a formally proven dynamical-systems attractor with an associated stability proof.)

With this notation fixed, we turn to the two derivations.

A.2 Quantifying the damage from linear erasure

Section 3.3 reports that preservation error degrades monotonically with entanglement. Here we derive the exact form of that degradation at the level of the bottleneck representation, and state precisely what additional assumption is needed to connect it to output-level reconstruction error.

Consider an input x in which only feature A is active, at magnitude $m_A > 0$, so that $h = m_A w_A$. Rank-1 nullspace projection onto the orthogonal complement of $\hat{u} = w_B/\|w_B\|$ gives

$$h' = h - (h \cdot \hat{u})\,\hat{u}. \tag{7}$$

The removed component has norm $\|h \cdot \hat{u}\| = m_A \|w_A\|\,|\rho|$, using $w_A \cdot w_B = \|w_A\|\|w_B\|\rho$ directly from the definition of ρ. Since $\|h\| = m_A\|w_A\|$, the fraction of h's norm removed by the projection is exactly $|\rho|$, independent of m_A, $\|w_A\|$, or d. Because reconstruction error is a squared quantity, the more relevant statistic is the fraction of squared norm removed:

$$\frac{\|h \cdot \hat{u}\|^2}{\|h\|^2} = \rho^2. \tag{8}$$

At the level of the bottleneck activation, this is exact: linear erasure destroys a ρ^2 fraction of A's representational energy along the shared subspace, with no free parameters or approximations involved.

Carrying this result forward to the output-level preservation MSE reported in Table 1 is a separate step, since output error depends additionally on the decoder row u_A, not on ρ alone. Under the assumption that the removed component of h contributes to output error in proportion to its squared norm, which holds when u_A is not systematically anti-aligned with the removed direction, preservation error should scale approximately as ρ^2 above a floor set by interference from the network's other represented features. This prediction is directionally correct throughout our sweep, MSE increases monotonically with $|\rho|$ and is largest at $\rho = -1$, but it is not an exact quantitative fit: at $\rho = -0.5$, a pure ρ^2 interpolation between the $\rho = 0$ and $\rho = -1$ endpoints predicts a preservation MSE around 0.176, against an observed value of 0.144. We attribute the gap to the two factors the assumption sets aside, the fourteen other represented features contributing to the baseline error floor independently of this projection, and the decoder direction u_A being only approximately aligned with this picture in a network trained end to end rather than constructed to satisfy it exactly.

A.3 A sufficient condition for nonlinear separability

Section 7 states informally that a threshold-based rule can exploit sign information a linear projection destroys. Building on the bottleneck-level result above, we make this precise for the fully antipodal case, $\rho = -1$, where $w_B = -w_A$ exactly (assuming, as in our construction, $\|w_A\| = \|w_B\|$).

Because A and B never co-occur, every input activates at most one of the two features, and the resulting bottleneck activation is always a scalar multiple of w_A. Writing $\hat{w}_A = w_A/\|w_A\|$ and $z = h \cdot \hat{w}_A$: an input with only A active at magnitude m_A gives $z = m_A\|w_A\| > 0$, and an input with only B active at magnitude m_B gives $h = m_B w_B = -m_B w_A$, so $z = -m_B\|w_A\| < 0$. The sign of z therefore identifies which feature is active with certainty.

A rule that erases B perfectly while leaving A untouched is, restricted to this one-dimensional subspace, the function $f(z) = z$ for $z \geq 0$ and $f(z) = 0$ for $z < 0$, exactly $\mathrm{ReLU}(z)$ (Nair & Hinton, 2010). No linear function of z can implement this: any linear map restricted to this subspace has the form $L(z\hat{w}_A) = cz$, which is odd, $L(-z\hat{w}_A) = -L(z\hat{w}_A)$, and an odd function cannot map every negative input to zero while leaving positive inputs unchanged. This is a restatement, in this restricted setting, of the general fact that a nontrivial linear operator cannot vanish on a ray without vanishing on its opposite ray as well.

This is close to what the mirror solution discovers on its own. By rotating w'_B toward $-w'_A$, the network arranges for B-only inputs to produce a negative pre-activation and A-only inputs to produce a positive one, then relies on the network's own ReLU nonlinearity, already present in the architecture, to implement the thresholding rule above. The mechanism is not designed into the network; it is discovered by gradient descent as the kind of solution capable of satisfying the erasure objective without the linear obstruction quantified in Appendix A.2. We note one point that can otherwise read as a contradiction: what determines the sign of the preactivation on the shared antipodal axis is each decoder row's projection onto $\hat{w}_A$ specifically, not its full-vector cosine similarity to the other decoder row. In a bottleneck of dimension $d > 1$, w'_A and w'_B can each carry a component along $\hat{w}_A$ small enough to satisfy preservation and suppression on this one axis, while differing substantially in the $d-1$ orthogonal directions; this is exactly what allows $\cos(w'_A, w'_B) \to -1$ at the level of full vectors (Figure 2) without contradicting the sign argument above, which concerns only the shared-axis projection.

With the theoretical picture in place, the next appendix turns to the experimental procedures that generate and classify the data these results are checked against.

B Experimental Methodology

This appendix specifies the procedures the main text refers to but does not spell out in full: grid construction for the phase sweep, the compute-budget safeguard, the basin classifier, confidence interval computation, and the functional definition of erasure used under GELU. Every appendix from here onward reports results produced by one of these procedures.

B.1 Phase sweep grid construction

The 15-point entanglement grid used in §4 is built in two passes rather than sampled uniformly, since the region of interest is not known in advance.

Pass 1 (pilot). We first run the full excision procedure at 5 entanglement values evenly spaced across $\rho \in [-1, 0]$, using 20 seeds per value rather than the full 100. We fit the basin classifier described in §B.3 to this pilot data and identify the entanglement range across which the basin fraction changes by more than 10 percentage points between adjacent pilot points; this range is treated as the transition zone.

Pass 2 (densified grid). The final 15-point grid places 5 points evenly across the full range, matching the pilot points, and 10 additional points evenly spaced within the transition zone identified in Pass 1. All 15 points are then rerun at the full 100 seeds per condition; pilot-run data is discarded rather than mixed with the full run, since the two use different seed counts. For the sweep reported in §4.3, the identified transition zone placed the crossover near $\rho \approx -0.75$, and the 10 densified points concentrate between $\rho = -0.75$ and $\rho = -0.5$ accordingly.

B.2 Compute budget and the reduced-grid safeguard

Before committing to a full sweep at a new configuration, a new architecture in §6, for instance, we run a timed pilot of 5 seeds at a single entanglement condition and extrapolate the wall-clock cost of the full grid. If the extrapolated cost exceeds six hours, the grid is reduced from 15 to 7 points rather than run in full. The 7 retained points are chosen by taking indices $\{0, \lfloor n/8 \rfloor, \lfloor n/4 \rfloor, \lfloor 3n/8 \rfloor, \lfloor n/2 \rfloor, \lfloor 3n/4 \rfloor, n-1\}$ into the sorted 15-point grid ($n = 15$), which guarantees both endpoints are retained but is not a symmetric selection: for the specific 15-point grid used in this paper, it evaluates to $\rho \in \{0, -0.071, -0.214, -0.357, -0.5, -0.786, -1.0\}$, which samples the $\rho \in [-0.5, 0]$ half more densely than the $\rho \in [-1, -0.5]$ half, and correspondingly under-samples the region immediately around the crossover reported in §4.3 ($\rho \approx -0.75$). Seed count is never reduced as part of this safeguard, only the number of entanglement points, since seed count determines the width of the confidence intervals reported in §4.3. This safeguard was triggered for both the depth experiment (§6.1) and the GELU experiment (§6.2); both therefore share this asymmetric coverage, which we note as a limitation of the resulting robustness checks rather than a deliberate design choice.

B.3 BASIN CLASSIFIER

Basin labels are assigned by fitting a two-component Gaussian mixture model, using the expectation-maximization algorithm (Dempster et al., 1977) as implemented in scikit-learn (Pedregosa et al., 2011), to the pooled $(\cos(w'_A, w'_B), b_B)$ points from a completed sweep, using full covariance matrices, and ranking the two fitted components by mean output bias. The component with the more negative mean bias is labeled shadow; the other is labeled mirror. We rank by bias rather than cosine similarity because, across the full sweep, bias is the more reliably separating axis: mean cosine similarity within a basin varies substantially with entanglement (§4.2), while the bias gap between basins is comparatively stable. Each run is assigned the label of its higher-posterior-probability component; runs used for the causal validation experiments in §5 are additionally required to have posterior probability exceeding 0.9 for their assigned basin. Under GELU (§6.2), where the two-component assumption does not hold, we instead select the number of mixture components by Bayesian information criterion (Schwarz, 1978) over $k \in \{2, 3, 4\}$ and report the winning k directly, rather than forcing a two-basin label onto data that may not support one.

B.4 CONFIDENCE INTERVALS

Basin fractions reported in §4.3 are binomial proportions out of 100 seeds, for which the normal approximation is unreliable near 0 or 1. We use the Wilson score interval (Wilson, 1927) throughout:

$$\hat{p} \pm \frac{z\sqrt{\hat{p}(1-\hat{p})/n + z^2/4n^2}}{1 + z^2/n}, \qquad \text{centered at } \frac{\hat{p} + z^2/2n}{1 + z^2/n}, \tag{9}$$

with $z = 1.96$ for the 95% intervals shown in Figure 4, computed independently at each of the 15 swept entanglement values.

B.5 FUNCTIONAL ERASURE UNDER GELU

GELU (Hendrycks & Gimpel, 2016) has no region that is identically zero, so the ReLU-based notion of erasure, output magnitude falling to exactly zero, does not directly apply. We instead define erasure functionally: at a given point in training, feature B is considered erased if the effect size of a causal patch restoring its pre-excision parameters, measured as the change in B's output, exceeds the effect size of an identical patch applied to a randomly chosen control feature not involved in the antipodal pair. This ties the GELU erasure criterion to the same causal-intervention logic used in §5, rather than introducing a separate, activation-value-based definition specific to one experiment. We note that this makes the GELU criterion different in kind from the ReLU one: it classifies a run by whether a patch has an effect, rather than by a property of the unpatched forward pass alone. This is not the same failure mode documented in Appendix E, where an unfrozen encoder could collapse B's encoder column and genuinely delete the representation; that failure mode is unavailable here because the encoder remains frozen throughout the GELU experiment exactly as in the ReLU case (§4.1), so there is no representation for the patch to fail to find. The criterion is nonetheless a different kind of measurement than a direct magnitude threshold, and we did not construct a magnitude-based GELU analogue, for instance a threshold on how small the pre-excision-scaled output gets, to cross-check against it; we treat this as a limitation of the GELU analysis rather than as a settled methodological choice. Appendix D.3 discusses this in more detail when weighing why GELU's basin structure looks different from ReLU's.

B.6 FULL LINEAR BASELINE SWEEP

The main text illustrates the linear baseline of §3.3 with three representative entanglement values, $\rho \in \{0, -0.5, -1\}$, since those three points already make the qualitative trend visible without crowding the main-text table. Table 3 gives the same measurement at all 15 entanglement values used in this baseline's own sweep, confirming that the trend visible in the three-point summary holds continuously across the full range rather than being an artifact of which points were chosen for the main text. This sweep's own densification, concentrated between $\rho = -0.5$ and $\rho = -0.75$, is separate from the phase-sweep grid of Appendix B.1, which is densified between $\rho = -0.9$ and $\rho = -0.6$; the two experiments use independent grids, each chosen for its own transition region.

Table 3: Linear baseline preservation MSE for A, full 15-point sweep, mean over 5 seeds.

ρ	**Preservation MSE** (A)
0.000	0.0982
-0.250	0.1144
-0.500	0.1440
-0.523	0.1481
-0.545	0.1526
-0.568	0.1577
-0.591	0.1639
-0.614	0.1708
-0.636	0.1788
-0.659	0.1886
-0.682	0.2006
-0.705	0.2161
-0.727	0.2368
-0.750	0.2622
-1.000	0.4112

Table 4: BIC scores for candidate component counts, GELU basin classifier

Components k	**2**	**3**	**4**
BIC	-4946.5	-4961.8	-5096.1

With the grid, classifier, and confidence intervals now specified, the next appendix uses them to give the full evidence behind the bifurcation phenomenon reported in §4.

C Supporting Evidence for the Bifurcation

Section 4 classifies every run by the procedure of Appendix B.3 and summarizes the result as two basins with a sharp crossover. This appendix gives the diagnostics behind that summary, first checking the classifier's labels directly against the underlying geometry, then examining what the classifier finds under GELU, where the two-basin picture does not hold as cleanly.

C.1 Basin geometry at full antipodal entanglement

The claim in §4.2 that mirror and shadow differ in degree of decoder rotation, rather than in whether rotation occurs at all, rests on inspecting the two basins directly rather than trusting the classifier's labels alone. The pre-excision decoder cosine similarity at $\rho = -1$, measured immediately before the excision phase begins, is -0.800; a strictly bias-only mechanism, with the decoder direction left at this pre-excision value, would predict both basins remain near -0.800 after excision. Neither does. The mirror-labeled cluster moves further negative, to a mean final decoder cosine similarity of -0.933, close to full reflection; the shadow-labeled cluster moves substantially the other way, to a mean of -0.401, a large shift toward orthogonality rather than away from it. Both movements are inconsistent with a bias-only account, which is the direct evidence behind describing shadow as a partial rather than absent rotation, not merely a difference in bias.

C.2 Basin classifier diagnostics under GELU

Section 6.2 reports that a Bayesian information criterion comparison favors four mixture components under GELU rather than two. Table 4 gives the compared BIC scores; Table 5 gives the fitted mean of each of the four components in $(\cos(w'_A, w'_B), b_B)$ space.

Table 5: Fitted component means, four-component GELU classifier

Component	$\cos(w'_A, w'_B)$	b_B
1	-0.733	0.0040
2	0.005	0.0070
3	0.079	0.0005
4	-0.999	0.0040

Table 6: Suppression locus distribution, depth-2 model, pooled across the full sweep

Locus	**Count**	**Fraction**
Output-layer dominant	637	0.910
Hidden-layer dominant	63	0.090
Split	0	0

Two features of this fit are worth noting. One component (component 4) sits almost exactly at full reflection, essentially a sharper version of the ReLU mirror solution. Output bias, meanwhile, is close to zero in every component, not only the ones with near-zero cosine similarity; this is the basis for the claim in §6.2 that GELU's soft negative region changes which suppression strategies are available, since bias suppression, the dominant mechanism for the ReLU shadow solution, does not appear to be strongly used under GELU at all. Appendix D.3 returns to this observation when weighing possible explanations for the GELU structure.

The classifier diagnostics above establish that the mirror/shadow distinction is real and visible in the raw geometry, not only in the labels the classifier assigns. The next appendix turns from this evidence to the robustness checks of §6 in full.

D Robustness: Full Results and Ruling Out Alternatives

Section 6 summarizes two robustness checks, an additional hidden layer and a switch from ReLU to GELU. This appendix gives the full results behind both, and, since the GELU result raised a genuine question rather than a clean confirmation, weighs the candidate explanations for it before settling on the one the main text reports.

D.1 Depth: full suppression-locus breakdown

Section 6.1 reports that suppression under a depth-2 model concentrates mainly in the output layer. We classify suppression locus per seed by comparing the bias shift at the output layer against the corresponding shift at the hidden layer, each normalized by that layer's own activation scale, and label a run as output-dominant, hidden-dominant, or split according to which comparison is larger. Table 6 gives the full distribution across the 7-point reduced grid used for this experiment (Appendix B.2), pooled across all seeds and entanglement conditions.

D.2 GELU: full basin distribution

The BIC comparison in Appendix C.2 establishes that four components fit the GELU data better than two, and Table 5 gives each component's location in $(\cos(w'_A, w'_B), b_B)$ space. What that comparison does not show is how many seeds actually fall into each of the four components, which matters for judging whether the four-component structure reflects a genuinely different population of outcomes or a small number of seeds pulling the fit away from a simpler two-component picture. Table 7 gives that breakdown, pooled across the same 7-point reduced grid used for the depth experiment above.

Table 7: Basin distribution under the four-component GELU classifier, pooled across the full sweep

Component	Count	Fraction
1 (partial rotation, low bias)	152	0.217
2 (near-orthogonal, low bias)	22	0.031
3 (near-orthogonal, low bias)	422	0.603
4 (near-full reflection, low bias)	104	0.149

D.3 Candidate explanations for the GELU basin structure

Three explanations are worth distinguishing before treating the four-component result as a genuine property of GELU rather than an artifact of how it was measured.

Measurement artifact. The GELU sweep uses a reduced 7-point grid rather than the full 15 (Appendix B.2), and this alone could plausibly produce a less clean fit simply from pooling over fewer entanglement conditions. We consider this unlikely to be the primary explanation, since the ReLU classifier was validated on data from the same style of reduced grid elsewhere without a comparable increase in component count, but we have not run the GELU sweep at the full 15 points to rule this out directly, and treat it as an open question rather than a settled one.

A genuinely different mechanism family. The component means in Table 5 show output bias staying close to zero across all four components, in contrast to the clearly negative bias that characterizes both ReLU basins. If GELU's soft negative region provides enough suppression on its own, without a strongly negative bias, bias suppression may simply not be a mechanism GELU-based excision needs to use. Under this account, the four components reflect degrees of decoder rotation on a single geometric axis, without the second, bias-driven axis that separates ReLU's two basins.

Reduced statistical power. Because the classifier is fit on data from a 7-point grid rather than 15, we cannot fully distinguish this from the measurement-artifact explanation above without a matched full-grid rerun.

Of the three, we find the second, a genuinely different mechanism family rather than noisier measurement of the same one, the most consistent with the evidence in Table 5, and it is the explanation reported in §6.2. We do not consider it established beyond the first explanation with the evidence currently in hand, and flag the full-grid rerun described above as the direct way to settle it.

Both robustness checks in this appendix rely on the frozen-encoder procedure specified in §4.1. The next appendix documents why that procedure is necessary, by reporting what happens without it.

E The Encoder-Freezing Correction

The depth and GELU results in Appendix D, and every main-text result from §4 onward, depend on holding the encoder W_E fixed during excision. This appendix documents why that choice matters, by reporting what an earlier, unfrozen version of the procedure produced.

E.1 Encoder collapse under unfrozen training

In an early version of the excision procedure, all parameters, including the encoder, were left trainable. Under this setup, the excision loss $\mathbb{E}_x \hat{x}_B^2$ can be minimized not only by suppressing B at readout, but also by shrinking B's encoder column w_B toward zero, which reduces B's contribution to the bottleneck activation directly and satisfies the objective regardless of what the decoder does. We observed exactly this: across seeds at $\rho = -1$, the norm of w_B collapsed from its initialization scale to approximately 0.002, roughly two orders of magnitude smaller than its starting value, while training loss decreased normally throughout.

E.2 Consequences for causal validation

This collapse is not an error in optimization; the network correctly minimizes the loss it was given. The problem is that it answers a different question than the one we intend to ask. Once w_B has collapsed, B's information is absent from the bottleneck activation h before it reaches the decoder, so there is nothing left for the decoder or output bias to suppress, and nothing left for a causal patch to recover. Applying the causal validation procedure of §5 to a model trained this way produces a patch with no measurable effect, not because the patch is incorrectly targeted, but because the quantity being patched no longer carries the information the patch is meant to restore. We confirmed this directly: dose-response curves computed on unfrozen-encoder models showed zero readout for B at every value of α, including $\alpha = 1$, in contrast to the smooth, substantial recovery reported in §5.2 for the frozen-encoder models used throughout the rest of this paper.

E.3 The correction

Freezing W_E throughout the excision phase, by excluding it from the optimizer's parameter set rather than merely zeroing its gradient after the fact, removes this route entirely. With W_E fixed, B's representation in h is guaranteed to persist through excision regardless of what the decoder and output bias learn to do, since nothing in the model can alter it. This is what allows the causal validation results in §5 to be read as evidence about suppression specifically, rather than as evidence that the network learned to stop representing B altogether. All results reported in the main text use the frozen-encoder procedure; the unfrozen-encoder results in this appendix are reported only to document why the correction is necessary.

E.4 Why this boundary condition might not hold at scale

Most practical unlearning setups, including full fine-tuning of a language model, do not hold earlier layers fixed the way we do here, which raises a question our results do not answer: why would suppression neurons appear in practice (Yang et al., 2026) if genuine deletion is available whenever upstream parameters are free to move? We do not know the answer, but two considerations suggest deletion may be less available in practice than the encoder-freezing comparison implies. First, our antipodal pair is isolated and load-bearing for nothing else, so collapsing B's encoder column costs the model nothing beyond B itself; in a language model, the corresponding parameters are almost certainly shared across many other computations, so a change large enough to delete one capability may be locally costly to many others, disincentivizing full deletion. Second, deletion in our setting requires the optimizer to find and traverse a norm-collapse direction which suppression does not, and it is not obvious this direction remains as cheap to find in a much higher-dimensional, more entangled parameter space. Both of these are hypotheses about why the boundary condition we control for here might not hold at scale, not conclusions our experiments establish; distinguishing between them, or finding that neither holds, is a natural target for follow-up work.

The correction above concerns how a fixed geometry is trained against, not where that geometry comes from in the first place. The next appendix addresses the latter question directly, checking whether antipodal geometry arises without any intervention on our part.

F Natural Emergence of Antipodal Geometry: Full Sweep

Section 4.4 reports that antipodal geometry emerges under ordinary training, without any direct geometric intervention, and summarizes the resulting sweep in a single sentence. This appendix gives the full grid behind that claim.

F.1 Sweep design

We train models from scratch with W_E free throughout, in contrast to every excision-phase experiment elsewhere in this paper, in which W_E is fixed, since this sweep is specifically about how the encoder geometry itself arises. We vary three quantities independently: sparsity $S \in \{0.5, 0.7, 0.8, 0.9, 0.95, 0.99\}$, bottleneck dimension $d \in \{2, 4, 8, 16\}$, and the strength of anti-correlation imposed between features A and B in the training data, ranging from 0.00 (no im-

Table 8: Fraction of converged seeds reaching antipodal geometry, marginalized over sparsity and bottleneck dimension

Anti-corr. strength	**0.00**	**0.25**	**0.50**	**0.75**	**1.00**
Fraction antipodal	0.068	0.296	0.351	0.405	0.562

posed anti-correlation) to 1.00 (strict mutual exclusivity). We use 10 seeds per grid cell, for $6 \times 4 \times 5 \times 10 = 1200$ runs total. For each run, we record whether training converged and, if so, the final encoder cosine similarity $\cos(w_A, w_B)$; a run is counted as having reached antipodal geometry if this value falls below -0.9.

F.2 Convergence

Not every run in this grid reaches a stable geometry; we restrict the results below to runs that converge, since a non-converged run has not yet reached any stable geometry and including it would conflate optimization failure with a genuine absence of antipodal structure. A run is considered converged if its training loss plateaus, defined as a relative decrease of less than 1% in mean MSE loss between the penultimate 10% of training steps and the final 10% of training steps; runs still reducing loss by more than 1% over this window at the end of training are excluded from both the numerator and denominator of every reported fraction. The runs excluded on this basis are concentrated at high sparsity ($S \geq 0.9$) combined with small bottleneck dimension, where the training signal for any individual feature pair is comparatively weak. Because the number of converged seeds varies by grid cell, Figure 7 reports the converged count alongside each cell's fraction directly, so a fraction computed from a small surviving N is legible as such rather than presented on equal footing with one computed from the full 10 seeds.

F.3 Results

Figure 7 shows the fraction of converged seeds reaching antipodal geometry across the full grid. Aggregated across sparsity and dimension, the pattern is monotonic in anti-correlation strength: the fraction of converged seeds reaching $\cos(w_A, w_B) < -0.9$ rises from 6.8% with no imposed anti-correlation to 56.2% under strict mutual exclusivity (Table 8). This is the evidence behind the claim in §4.4 that antipodal geometry is not an artifact of the construction used in §4, but a structure training finds on its own once two features are prevented from co-occurring, consistent with the general account of superposition given by Elhage et al. (2022).

The full grid in Figure 7 additionally shows that this relationship is not uniform across sparsity and dimension. Imposed anti-correlation is doing most of the work: at anti_corr_prob $= 0$, the antipodal fraction stays low (at most 0.30, and 0.00 at the lowest sparsity value) even at the smallest bottleneck dimension, so sparsity and dimension alone, without any imposed anti-correlation, are not sufficient to reliably produce antipodal geometry. Once anti-correlation is imposed at all, low bottleneck dimension and moderate-to-high sparsity combine with it to produce near-certain antipodal emergence; at high sparsity combined with a wide bottleneck, by contrast, antipodal geometry stays rare even under strict mutual exclusivity, since a wide bottleneck reduces the pressure to share a subspace at all. We read this as anti-correlation strength being the primary driver, with sparsity and bottleneck dimension acting as secondary factors that amplify or dampen its effect, rather than as independently sufficient causes on their own.

The sweep above confirms the geometry itself is natural. The next appendix checks whether the mechanism built on top of that geometry, the mirror/shadow bifurcation, is specific to the autoencoder architecture used throughout the rest of this paper.

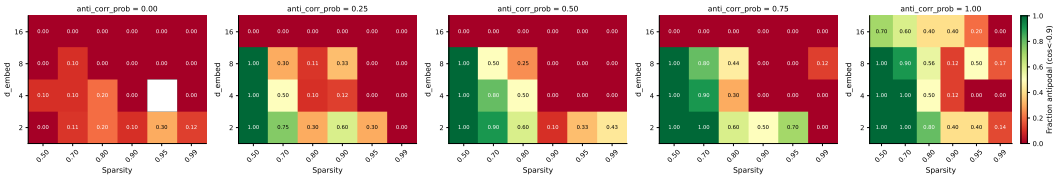


Figure 7: Fraction of seeds converging to antipodal geometry ($\cos(w_A, w_B) < -0.9$), as a function of sparsity, bottleneck dimension, and anti-correlation strength. Converged runs only; the number of converged seeds out of the nominal 10 per cell is printed below each fraction, since this varies by cell (§F.2) and a fraction computed from a small surviving count should be weighted accordingly.

G Transformer Scale Check: Model and Full Statistics

Section 6 reports a small-scale check for whether the mirror/shadow phenomenon appears outside the toy autoencoder setting. This appendix gives the exact model and the full statistics behind that check.

G.1 Model definition

We use a minimal decoder-only causal transformer (Vaswani et al., 2017), following notational conventions similar to Elhage et al. (2021), with two layers and two attention heads per layer. Token inputs are mapped through a learned embedding $E \in \mathbb{R}^{V \times d}$ and a learned positional embedding $P \in \mathbb{R}^{T \times d}$, where V is vocabulary size, T is sequence length, and d is model dimension; the residual stream at position t is initialized as $h_t^{(0)} = E_{x_t} + P_t$. Each layer applies standard causal self-attention followed by a two-layer MLP with GELU nonlinearity (Hendrycks & Gimpel, 2016), both with residual connections, and a causal mask preventing position t from attending to positions after t. The final hidden state at each position is mapped to logits by an unembedding matrix $U \in \mathbb{R}^{V \times d}$ with bias $b_U \in \mathbb{R}^V$: $\text{logits}_t = U h_t^{(2)} + b_U$.

The synthetic task uses two designated token types, A and B, constructed so that a single sequence never contains both, together with a shared vocabulary of filler tokens carrying no task-relevant signal, the direct analogue of the antipodal pair's mutual exclusivity in the main experiments, implemented at the level of tokens rather than continuous features. The model is pretrained with a standard next-token prediction objective using AdamW (Loshchilov & Hutter, 2019), then, analogously to the excision phase defined in §4.1, fine-tuned with the embedding, attention, and MLP sublayers of both blocks held fixed, leaving only the final unembedding weight U and bias b_U trainable, against an objective that penalizes the predicted probability of token type B.

G.2 Probing methodology

Before fine-tuning, we verify that the pretrained model's hidden states linearly encode which token type is present in a given sequence, since the causal patching argument that follows requires this representation to genuinely exist prior to any suppression. We fit a logistic regression probe, using the scikit-learn implementation (Pedregosa et al., 2011), on the final-layer hidden state at the last sequence position, holding the pretrained model fixed, and report probe test accuracy directly rather than assuming linear encoding holds.

G.3 Causal patch

The causal patch used in §6 follows the same forward-pass-override pattern defined in Appendix A.1 for the main experiments: b_U for the B token is replaced with its pre-fine-tuning value for a single forward pass, and the model's predicted probability for token B is compared before and after, with the underlying parameter never mutated.

G.4 FULL STATISTICS

Across 15 seeds, mean probe test accuracy was 0.905 or higher in every seed at the final sequence position, supporting the assumption in Appendix G.2 that the relevant representation is genuinely present before fine-tuning begins. A model-selection comparison over $k \in \{1, 2, 3\}$ mixture components on the fine-tuned (unembed cosine similarity, $(b_U)_B$) endpoints favors $k = 2$, with a BIC gap large enough to consider the comparison unambiguous by the same criterion used in Appendix C.2. We note this comparison rests on far fewer data points than the analogous comparisons elsewhere in this paper: a full-covariance 2-component Gaussian mixture on 2-dimensional endpoints has 11 free parameters, fit here to only 15 seeds, leaving very few effective degrees of freedom and a real risk that the model is close to interpolating the data outright. We report the comparison because $k = 2$ is what BIC selects, not because we consider it as well-supported as the corresponding $N = 100$ comparisons used for the main phase sweep, and would treat a $k = 2$ finding at this sample size as suggestive rather than conclusive on its own. Restoring the pre-fine-tuning unembedding bias for token B increases the model's mean predicted probability for that token from 0.0031 to 0.0035; the increase is small in absolute terms but consistent in direction across all 15 seeds (Wilcoxon signed-rank test, $p < 0.001$) (Wilcoxon, 1945), computed with SciPy (Virtanen et al., 2020), the basis for the claim in §6 that restoring the relevant bias measurably increases the suppressed token's predicted probability. We note two limitations of this result that do not apply to the corresponding autoencoder patches. First, because softmax probability is strictly monotonic in the bias of the token being restored, adding back a positive $\Delta b_U > 0$ algebraically guarantees $P(B)$ increases; the consistent direction and significance of the effect are therefore a necessary consequence of restoring a positive bias shift, not by themselves evidence that the patch reactivates a suppressed representation, in the way sufficiency and specificity independently establish for the autoencoder patches in §5.3. What is informative is the size of the increase relative to a matched negative control on an unrelated vocabulary token, which we did not run for this experiment; we report the raw effect size for that reason and do not claim it has been shown specific to token B in the way the main experiments' 20-trial negative controls establish. Second, an absolute probability of 0.0035 is small enough that we do not know whether it changes the model's top-1 generation behavior at any sequence position; we report the probability shift directly rather than a generation-level recovery metric, and consider whether the patch changes actual generated output an open question this appendix does not answer. We apply this single bias-restoration patch uniformly across all 15 seeds rather than splitting by the two components the BIC comparison identifies. This mirrors the toy autoencoder's shadow-solution patch but not its mirror-solution patch, so to the extent any of these 15 seeds are mirror-like in the same sense as §4.2, a bias patch alone may understate their true recoverability; we did not have a validated mirror-style patch for this architecture and leave a component-specific patching analysis of the transformer check to future work.

G.5 SCOPE

This experiment is a small-scale, seed-limited ($n = 15$) qualitative check for whether the mirror/shadow phenomenon generalizes beyond the toy superposition model to a minimal transformer with comparable but not identical training dynamics. It is not a claim of generality to production-scale language models, which differ in scale, tokenization, training data, and architecture in ways this experiment does not address. We also note a narrower scope limitation: with embeddings, attention, and MLP sublayers all held fixed during fine-tuning, this check evaluates suppression at a linear readout over a frozen representation, the direct transformer analogue of the autoencoder's frozen-encoder excision, rather than testing whether the same bifurcation arises when earlier, representation-forming layers are also free to adapt. We report it as motivation for future work at greater scale, and at greater architectural scope, rather than as evidence that the phenomenon holds under full end-to-end fine-tuning.

The empirical evidence assembled across Appendices C through G establishes that the mirror/shadow bifurcation is real, robust to depth and nonlinearity, and not confined to the autoencoder setting. What none of it explains is why entanglement in particular governs which basin dominates. The closing appendix takes up that question directly, as an open direction rather than a settled result.

H TOWARD A THEORETICAL ACCOUNT OF THE MIRROR/SHADOW BIFURCATION

The phase diagram in §4.3 establishes that the fraction of runs landing in the mirror basin rises sharply as entanglement approaches -1, but does not explain why entanglement should govern this choice at all. This appendix offers a candidate account. The reasoning here is exploratory: we have not derived or tested the specific prediction in §H.2, and present it to motivate future theoretical work rather than as a result of this paper.

H.1 A BASIN-OF-ATTRACTION FRAMING

At the start of excision, the decoder rows w'_A, w'_B sit at some initial cosine similarity determined by the fixed encoder geometry, since decoder and encoder are typically initialized close to each other in a well-trained autoencoder. From this starting point, gradient descent on the excision objective has, at minimum, two directions available: continue rotating w'_B toward $-w'_A$, which is the mirror route, or leave w'_B closer to where it started and instead drive the output bias b_B negative, which is the shadow route. (As in Appendix A.3, "toward $-w'_A$" describes the shared-axis projection that governs suppression on the antipodal axis specifically, not the full-vector direction of w'_B in the ambient d-dimensional space; see the note there for how this is consistent with $\cos(w'_A, w'_B) \to -1$ at the level of full vectors.) Which of the two is locally cheaper, in the sense of producing a larger immediate reduction in the excision loss for a small parameter step, plausibly depends on how much rotation the starting geometry already implies.

At high entanglement, w_A and w_B already point in nearly opposite directions, so a modest further rotation is enough to push B-only inputs into strongly negative pre-activation territory, at comparatively low cost to the reconstruction term, since the change interacts favorably with the already-antipodal encoder geometry. This favors the mirror route. At low entanglement, the same rotation would need to move w'_B substantially further to achieve the same separating effect, since little initial separation exists to build on, while a comparably sized step toward the shadow route may be cheaper. Under this account, entanglement does not force a discrete choice between two qualitatively different mechanisms so much as it shifts the relative local cost of the mirror route against the shadow route, both of which are always simultaneously available to the optimizer.

H.2 A TESTABLE PREDICTION

If this account is right, it predicts that the crossover point identified in §4.3 should be sensitive to the relative learning rates applied to the decoder weights versus the output bias, since the framing above turns on which route produces a larger loss reduction per unit of optimizer movement, and that ratio is set in part by the optimizer's own per-parameter step sizes, not by the loss landscape alone. We have not tested this. The direct check is to rerun the phase sweep of §4 at a deliberately mismatched decoder-to-bias learning rate ratio, for instance $\eta_{\text{dec}}/\eta_{\text{bias}} \in \{0.1, 1.0, 10.0\}$: if the crossover point shifts in the direction this account predicts, that supports it; if the crossover is largely insensitive to the ratio, the account is better replaced by an explanation grounded in the loss landscape's geometry alone, independent of optimizer step sizes. We note that this prediction is stated in terms of raw parameter-step size, which is the relevant notion under vanilla gradient descent but not under Adam-family optimizers (Kingma & Ba, 2015), including the AdamW (Loshchilov & Hutter, 2019) used throughout our experiments, since these normalize each coordinate's update by a running estimate of its own gradient scale and decouple weight decay from the gradient step itself. A learning-rate-ratio sweep under AdamW is therefore not a clean test of the Euclidean step-cost story as stated; disentangling a genuine loss-landscape effect from an artifact of Adam's per-coordinate normalization would need to be part of the same follow-up experiment, for instance by comparing the sweep under AdamW against the same sweep under plain SGD.

H.3 DIRECTIONS FOR FUTURE THEORETICAL WORK

A complete account would characterize the excision loss landscape's local curvature around plausible decoder configurations as a function of entanglement directly, rather than reasoning informally about relative step costs as we do above. We view this, together with the empirical check proposed in §H.2, as the most direct open theoretical question raised by this work, and one we think is better

pursued as its own dedicated analysis than compressed into an appendix of an empirically focused paper.

I Directions for Empirical Validation

§7 names three comparisons we consider the most direct next steps for testing whether this account generalizes beyond the setting we constructed it in. We have not run any of them; this appendix states each in more detail than main-text space allows.

Post-hoc nonlinear erasure. We motivate gradient-based excision by showing linear projection fails at $\rho = -1$ (§3.3), but we do not test whether post-hoc nonlinear erasure methods designed for exactly this failure mode, such as kernelized concept erasure (Ravfogel et al., 2022b) or kernelized rate-distortion maximization (Basu Roy Chowdhury et al., 2023), avoid the mirror/shadow bifurcation entirely by finding a transformation that genuinely removes B rather than suppressing it at readout. If one does, that would be evidence the bifurcation is specific to training-time optimization rather than to nonlinear erasure in general; if none does, that would suggest the bifurcation is a more fundamental consequence of forcing a nonlinear separation on this geometry, independent of whether the separation is learned during training or applied after the fact.

Alternative unlearning objectives. Our excision objective is a specific, bespoke choice, reconstruction loss on the retain set plus a squared-activation penalty on B, and it remains open whether the same bifurcation appears under objectives drawn directly from the unlearning literature, such as representation misdirection (Li et al., 2024) or a student-teacher distillation objective. A different objective could plausibly land in only one basin, or in neither, which would narrow the scope of what we report here considerably; conversely, finding the same bifurcation under an objective we did not design specifically to produce it would be stronger evidence that mirror and shadow are properties of the erasure problem itself rather than of our particular loss.

Validation on standard benchmarks. Every claim in this paper is made on synthetic data constructed to have the antipodal structure we study. We do not know whether the mirror/shadow distinction, or anything resembling it, is visible when the same excision procedure is applied to a standard unlearning benchmark such as TOFU (Maini et al., 2024) or Bias in Bios (De-Arteaga et al., 2019), where the relevant features are not hand-constructed to be antipodal and may not be separable from the rest of the representation in as clean a way. This is the comparison most directly relevant to practical impact, and also the one least likely to reproduce our results cleanly, since real feature geometries are unlikely to be as sharply antipodal as the setting we construct.