REPRESENTATIONAL CHANGE UNDER SELF-CONSUMING TRAINING: A SPARSE AUTOENCODER ANALYSIS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models are increasingly exposed to model-generated text, raising the possibility that future models will be trained recursively on the outputs of their predecessors. Prior work on such self-consuming training loops has largely focused on output quality and downstream performance, leaving their effects on internal representations poorly understood. We study how representations evolve across five generations of self-consumption and whether sparse autoencoders (SAEs) remain reliable under this shift. At each generation, we compare a lineage trained on synthetic data with a matched control trained on real data under the same training settings. We evaluate how well SAEs trained before self-consumption transfer to later generations, measuring reconstruction and sparsity across SAE widths, architectures, and model layers. We then fine-tune the original SAE on each generation and use sparse probing to test whether task-relevant information remains concentrated in a small number of latents or becomes more distributed across the representation. We find that zero-shot SAE performance degrades progressively under self-consumption, with growing reconstruction and sparsity gaps that vary substantially across layers. Fine-tuning largely restores reconstruction, but not feature-level usefulness: task-relevant information remains accessible in the full latent space while becoming less concentrated in individual features. These results show that self-consumption induces representational drift and that reconstruction fidelity alone is insufficient to establish that an adapted SAE remains equally useful for interpretation.

1 INTRODUCTION

Language models are increasingly trained on corpora that contain text generated by other language models. As high-quality human-written data becomes harder to scale (Villalobos et al., 2024), machine-generated text is accumulating rapidly on the web and even in scientific writing (Liang et al., 2024). Future models are therefore likely to be trained, at least in part, on the outputs of their predecessors. Repeating this process creates a *self-consuming loop* (Alemohammad et al., 2024), in which one model generates training data for the next, and the cycle continues across generations. Prior works show that such recursive training can progressively narrow the learned distribution and erode its tails, a phenomenon known as model collapse (Shumailov et al., 2023; 2024). However, these effects have primarily been studied through externally observable quantities such as likelihood, output diversity, and downstream performance (Refer Sec 2).

This leaves a complementary question largely unexplored: what happens to the model's internal representations under self-consumption? This question is particularly important for interpretability, since many methods operate directly on model internals (Alain & Bengio, 2016; Belinkov, 2022; Sharkey et al., 2025). Sparse autoencoders (SAEs) have become a widely used tool for decomposing dense activations into sparse, individually interpretable features (Bricken et al., 2023; Cunningham et al., 2024; Templeton et al., 2024; Gao et al., 2025). As models evolve through repeated training on synthetic data, however, it is unclear whether an SAE trained on an earlier model will continue to transfer, or whether the features recovered after adaptation retain the same usefulness. We therefore use SAEs as a diagnostic of representational change and ask two questions:

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.

1. **Does the original SAE still transfer?** SAEs often transfer between related models, such as base and chat models (Kissane et al., 2024; Lan et al., 2024; Gupte et al., 2026), but whether they survive a self-consuming loop is unknown. Progressive failure of a fixed SAE is evidence of representational drift.
2. **Does adapting the SAE restore its usefulness?** Fine-tuning the original SAE on each new generation may restore reconstruction, but good reconstruction does not guarantee that information remains localized in individually useful features.

We study these questions over five generations of self-consumption training using `Pythia-160M` (Biderman et al., 2023) and `Gemma-2-2B` (Gemma Team, 2024). For `Pythia-160M`, we evaluate multiple SAE widths and architectures; for `Gemma-2-2B`, we examine early, middle, and late layers. Each synthetic lineage is paired with a real-data control trained with the same token budget, same number of optimization steps, and same number of generations, isolating the effect of training on synthetic rather than real data.

Across these matched lineages, we observe a consistent progression. Zero-shot SAE transfer degrades under self-consumption: on `Pythia-160M`, synthetic–real gaps in reconstruction and sparsity grow across generations for all SAE widths and architectures, while on `Gemma-2-2B` the effect is strongly depth-dependent and largest at the late layer. Fine-tuning the SAE on each generation largely restores reconstruction, suggesting that much of this degradation reflects a mismatch between the original SAE and the shifted activation distribution. However, recovering reconstruction does not necessarily recover feature-level usefulness. On `Pythia-160M`, sparse probes using only the top few latents remain degraded after fine-tuning even when probes over the full SAE representation recover, indicating that task-relevant information persists but becomes less concentrated in individual latents. On `Gemma-2-2B`, these feature-level effects vary across layers and metrics. Taken together, these results show that reconstruction quality and feature-level usefulness can diverge under self-consumption. Evaluating whether an adapted SAE remains a reliable interpretability tool therefore requires feature-level criteria in addition to reconstruction fidelity.

The remainder of the paper is organized as follows: Sec. 2 reviews related work, Sec. 3 introduces the matched-lineage setup and SAE evaluation regimes, and Sec. 4 develops a linear-autoencoder analysis that separates the reconstruction cost of a stale dictionary from the intrinsic reconstructability of the shifted representation, and shows why reconstruction recovery need not imply recovery of feature localization. Sections 5–7 then present the experiments, results, and broader implications.

2 Related Work

Self-consuming generative models: Recursive training on model-generated data can cause distributional degradation, or model collapse (Shumailov et al., 2023; 2024; Alemohammad et al., 2024). This effect has been documented in image generators (Hataya et al., 2023; Martínez et al., 2023) and language models, where recursive training can reduce linguistic diversity and downstream performance (Briesch et al., 2023; Guo et al., 2024). Theoretical work characterizes when iterative retraining remains stable (Bertrand et al., 2024), how collapse alters scaling laws (Dohmatob et al., 2024), and how errors accumulate in regression and simplified language-model settings (Seddik et al., 2024). Proposed mitigations include accumulating synthetic data alongside real data rather than replacing it (Gerstgrasser et al., 2024; Kazdan et al., 2024), filtering or verifying synthetic samples (Feng et al., 2024), and introducing self-correction into the generation loop (Gillman et al., 2024). Existing work primarily characterizes changes in the generated distribution or downstream behavior; we instead examine how self-consumption changes internal representations and the features recovered from them.

Sparse autoencoders: SAEs are motivated by the superposition hypothesis, under which models may represent more features than they have available dimensions (Elhage et al., 2022; Olshausen & Field, 1997). They decompose dense activations into an overcomplete sparse dictionary whose features are often individually interpretable (Bricken et al., 2023; Cunningham et al., 2024). SAEs have since been scaled to language models (Templeton et al., 2024; Gao et al., 2025) and released in open feature suites (Lieberum et al., 2024). Several variants improve the sparsity–reconstruction tradeoff, including gated (Rajamanoharan et al., 2024a), TopK (Gao et al., 2025), BatchTopK (Bussmann et al., 2024), JumpReLU (Rajamanoharan et al., 2024b), and Matryoshka (Bussmann et al., 2025)

SAEs, as well as end-to-end training objectives (Braun et al., 2024). SAE features have also been used for circuit discovery (Marks et al., 2025) and automated interpretation (Bills et al., 2023; Paulo et al., 2024). Most of this literature evaluates SAEs on a fixed underlying model; our setting instead asks what happens when the model itself evolves across generations.

Interpretability across changing models: Recent work suggests that some interpretability structure can persist across model changes. SAEs trained on base models can transfer to chat-tuned variants (Kissane et al., 2024; Gupte et al., 2025), feature spaces can be partially shared across language models (Lan et al., 2024), and SAE representations can remain stable under pruning (Gupte et al., 2026). Crosscoders explicitly learn joint dictionaries for comparing related models (Lindsey et al., 2024; Minder et al., 2025), while mechanistic studies of fine-tuning find substantial reuse and refinement of existing mechanisms (Jain et al., 2024; Prakash et al., 2024). More general representation-similarity methods such as SVCCA and CKA compare activations across models or training runs (Raghu et al., 2017; Kornblith et al., 2019). These settings typically examine a single transition between models. We instead study a sequence of recursive updates and ask whether SAE transfer degrades progressively as those changes accumulate.

Evaluating SAEs beyond reconstruction: Reconstruction fidelity alone does not determine whether an SAE has learned useful or interpretable features. Individual features can split or absorb one another (Chanin et al., 2024), reconstruction error can contain structured "dark matter" (Engels et al., 2024), different training seeds can produce different feature dictionaries (Paulo & Belrose, 2025), and SAEs trained on randomly initialized transformers can perform surprisingly well on some standard metrics (Heap et al., 2025). These limitations have motivated evaluations based on supervised dictionaries (Makelov et al., 2024), sparse probing (Gurnee et al., 2023; Kantamneni et al., 2025), steering benchmarks (Wu et al., 2025), and broader evaluation suites such as SAEBench (Karvonen et al., 2025). We adopt this broader perspective to distinguish recovery of reconstruction from recovery of feature-level usefulness.

Linear autoencoders and spectral perturbation. Our analysis in Sec. 4 builds on the equivalence between optimal linear autoencoders and principal component analysis (Bourlard & Kamp, 1988; Baldi & Hornik, 1989; Kunin et al., 2019). We combine this connection with eigenspace perturbation bounds (Davis & Kahan, 1970) and finite-sample covariance concentration (Vershynin, 2018) to analyze how repeated shifts in the activation covariance make a fixed dictionary increasingly mismatched to the evolving representation.

3 Problem Setup

3.1 Self-Consuming Training and a Matched Real-Data Control

Let $p^{(0)}$ be a language model pretrained on real data, and let $\mathcal{A}(p; \mathcal{D})$ denote continued next-token-prediction training of model p on dataset $\mathcal{D}$ with a fixed optimization procedure and token budget. A self-consuming loop runs for T generations starting from $p^{(0)}_{\text{syn}} = p^{(0)}$. At generation $t \in \{1, \dots, T\}$, the previous model generates a synthetic corpus on which it is then trained:

$$\tilde{\mathcal{D}}^{(t)} = \{x_j^{(t)}\}_{j=1}^{n}, \quad x_j^{(t)} \overset{\text{i.i.d.}}{\sim} p_{\text{syn}}^{(t-1)}, \qquad p_{\text{syn}}^{(t)} = \mathcal{A}\big(p_{\text{syn}}^{(t-1)}; \tilde{\mathcal{D}}^{(t)}\big). \tag{1}$$

Comparing generation t with generation 0 would conflate the effect of synthetic data with the change caused by any continued training. We therefore construct a control lineage that starts from the same model and uses the same token budget and optimization procedure, but is trained on a fresh slice of real data at every generation:

$$p_{\text{real}}^{(t)} = \mathcal{A}\big(p_{\text{real}}^{(t-1)}; \mathcal{D}_{\text{real}}^{(t)}\big), \qquad p_{\text{real}}^{(0)} = p_{\text{syn}}^{(0)} = p^{(0)}. \tag{2}$$

The two lineages thus differ only in whether the continued-training data are generated by the previous model or drawn from real text.

3.2 Zero-Shot and Fine-tuned SAEs

Even with this control, a synthetic–real gap can arise for two different reasons: self-consumption may shift the activation distribution away from the one the SAE was trained on, leaving a stale

dictionary, or it may change the geometry of the representation the dictionary is trying to describe. We separate these with two evaluation regimes. Let $S^{(0)}$ be an SAE trained on activations of $p^{(0)}$, and let $h_\ell^{(t)}$ denote activations of $p_\ell^{(t)}$ for lineage $\ell \in \{\text{syn}, \text{real}\}$.

Zero-shot (frozen) SAE. We apply $S^{(0)}$ unchanged to every generation, $S_\ell^{(t)} = S^{(0)}$. This measures zero-shot transfer of the original dictionary as the representation changes.

Fine-tuned SAE. We initialize from $S^{(0)}$ and continue training on activations of the target model, $S_\ell^{(t)} = \text{SAEFT}\big(S^{(0)}; h_\ell^{(t)}\big)$, separately for each lineage and generation. The dictionary is adapted, not retrained from scratch.

The two regimes coincide at generation 0. If fine-tuning closes the synthetic–real gap, the damage lies in a stale dictionary; if the gap persists, self-consumption has changed the representation in a way that adapting the dictionary cannot undo.

3.3 Measuring the Effect of Self-Consumption

Let $m(p, S)$ denote an SAE evaluation metric computed with SAE S on activations of model p. At each generation t, we compare the synthetic lineage with the generation-matched control:

$$\text{Damage}_m^{(t)} = s_m \Big[m\big(p_{\text{real}}^{(t)}, S_{\text{real}}^{(t)}\big) - m\big(p_{\text{syn}}^{(t)}, S_{\text{syn}}^{(t)}\big) \Big], \tag{3}$$

where $s_m = +1$ if larger values of m are better and $s_m = -1$ otherwise. Positive damage therefore always means that the synthetic lineage performs worse than matched continued training on real data, and zero means no degradation attributable to self-consumption. Since both lineages start from the same model and SAE, damage is zero at generation 0 by construction.

4 Analysis of SAE Staleness Under Representational Drift

Our experiments distinguish two phenomena: a fixed SAE can become poorly matched to an evolving representation, while an SAE adapted to that representation can still reconstruct it well. We formalize this distinction using a linear autoencoder as an analytically tractable proxy for an SAE. The goal is to isolate when reconstruction degradation reflects a stale dictionary, and why recovering reconstruction need not recover feature interpretability. Full derivations are given in Appendix B.

Linear autoencoders recover a principal subspace. Let $h_\ell^{(t)} \in \mathbb{R}^d$ denote centered activations at generation t from lineage $\ell \in \{\text{syn}, \text{real}\}$, with covariance $\Sigma_\ell^{(t)}$. Consider a rank-k linear autoencoder with orthonormal decoder W and projection $P = WW^\top$. Its reconstruction error is

$$\mathcal{L}(P; \Sigma) = \mathbb{E}\|h - Ph\|_2^2 = \text{tr}[(I - P)\Sigma]\,. \tag{4}$$

The optimal rank-k projection $P_\ell^{(t)}$ is the projection onto the top-k eigenspace of $\Sigma_\ell^{(t)}$. Thus, changes in the activation covariance can change the subspace that is optimal for reconstruction.

Representational drift creates a staleness penalty. Let $P^{(0)}$ denote the projection learned at generation 0. The zero-shot linear SAE uses this fixed projection at every later generation,

$$\mathcal{L}_\ell^{\text{zs}}(t) = \mathcal{L}\Big(P^{(0)}; \Sigma_\ell^{(t)}\Big)\,, \tag{5}$$

whereas an ideally adapted SAE uses the current optimal projection,

$$\mathcal{L}_\ell^{\text{adapt}}(t) = \mathcal{L}\Big(P_\ell^{(t)}; \Sigma_\ell^{(t)}\Big)\,. \tag{6}$$

Their difference,

$$S_\ell(t) := \mathcal{L}_\ell^{\text{zs}}(t) - \mathcal{L}_\ell^{\text{adapt}}(t) = \text{tr}\left[\Big(P_\ell^{(t)} - P^{(0)}\Big)\,\Sigma_\ell^{(t)}\right] \geq 0, \tag{7}$$

is the reconstruction cost attributable specifically to using a stale generation-0 dictionary.

This penalty is controlled by the misalignment between the original and current principal subspaces. Let $\Theta_\ell^{(t)}$ denote their principal angles and let

$$\gamma_\ell^{(t)} = \lambda_k\left(\Sigma_\ell^{(t)}\right) - \lambda_{k+1}\left(\Sigma_\ell^{(t)}\right)$$

be the relevant eigengap. Then

$$\gamma_\ell^{(t)} \| \sin \Theta_\ell^{(t)} \|_F^2 \le S_\ell(t) \le \left(\lambda_1\left(\Sigma_\ell^{(t)}\right) - \lambda_d\left(\Sigma_\ell^{(t)}\right)\right) \| \sin \Theta_\ell^{(t)} \|_F^2. \tag{8}$$

Hence a frozen dictionary incurs increasing reconstruction cost when the reconstruction-relevant subspace moves away from the one on which it was trained. Standard eigenspace perturbation results further relate this misalignment to cumulative covariance shift; we give the corresponding finite-sample bounds in Appendix B.

Why fine-tuning closes the synthetic–real gap. The matched control makes this decomposition particularly useful. Define reconstruction damage under zero-shot and adapted evaluation as

$$D_{\mathrm{zs}}(t) = \mathcal{L}_{\mathrm{syn}}^{\mathrm{zs}}(t) - \mathcal{L}_{\mathrm{real}}^{\mathrm{zs}}(t), \tag{9}$$

$$D_{\mathrm{adapt}}(t) = \mathcal{L}_{\mathrm{syn}}^{\mathrm{adapt}}(t) - \mathcal{L}_{\mathrm{real}}^{\mathrm{adapt}}(t). \tag{10}$$

Then

$$D_{\mathrm{zs}}(t) - D_{\mathrm{adapt}}(t) = S_{\mathrm{syn}}(t) - S_{\mathrm{real}}(t). \tag{11}$$

Thus, if self-consumption induces greater subspace drift than fine-tuning on real data, the synthetic lineage acquires a larger staleness penalty, and adapting the SAE removes this component of the synthetic–real gap precisely. This is the pattern observed in our reconstruction experiments.

Reconstruction recovery need not imply feature recovery. Crucially, reconstruction constrains a subspace more strongly than it constrains the individual coordinates within that subspace. Suppose two representations of the same k-dimensional subspace differ by an orthogonal change of basis, $W' = WQ$. Then

$$W'W'^\top = WQQ^\top W^\top = WW^\top,$$

so their reconstruction is identical. Their latent coordinates, however, satisfy $z' = Q^\top z$ and can distribute information very differently across individual dimensions. A linear probe using the full latent vector is invariant to this transformation: for every probe $a^\top z$ there is a probe $(Q^\top a)^\top z'$ with identical predictions. In contrast, information that was concentrated in one or a few coordinates can be spread across many coordinates after the change of basis.

This gives a simple explanation for why reconstruction and sparse localization can diverge: adaptation can recover a good representation of the relevant subspace without requiring task-relevant information to remain concentrated in the same small number of SAE latents. The argument predicts the possibility, rather than the necessity, of feature-level degradation, consistent with the strong effect in Pythia and the more heterogeneous behavior in Gemma.

5 Experimental Setup

Self-consuming lineages. We study five generations of synthetic and matched real-data training using `Pythia-160M` (Biderman et al., 2023) and `Gemma-2-2B` (Gemma Team, 2024). Starting from the same pretrained checkpoint, each synthetic generation is trained exclusively on text generated by the previous synthetic model, while the real-data lineage is trained on a fresh slice of `monology/pile-uncopyrighted`. Both lineages use the same training budget and optimization setup, isolating the effect of synthetic rather than continued training. We analyze Pythia layer 8 of 12, matching SAEBench, and Gemma layers 5, 12, and 19 to test how representational change varies with depth. Full training and sampling details are provided in Appendix A.1.

SAE configurations. For `Pythia-160M`, we vary both SAE width and architecture. Our width sweep uses BatchTopK (Bussmann et al., 2024) SAEs with widths $\{2^{12}, 2^{14}, 2^{16}\}$, while our architecture sweep fixes the width at 2^{14} and compares BatchTopK, JumpReLU (Rajamanoharan et al., 2024b), and Matryoshka-BatchTopK (Bussmann et al., 2025). For `Gemma-2-2B`, we use width-2^{14} JumpReLU SAEs at each evaluated layer. Every SAE is evaluated under both regimes from Sec. 3.2. For fine-tuning, we use three seeds and report their mean and spread.

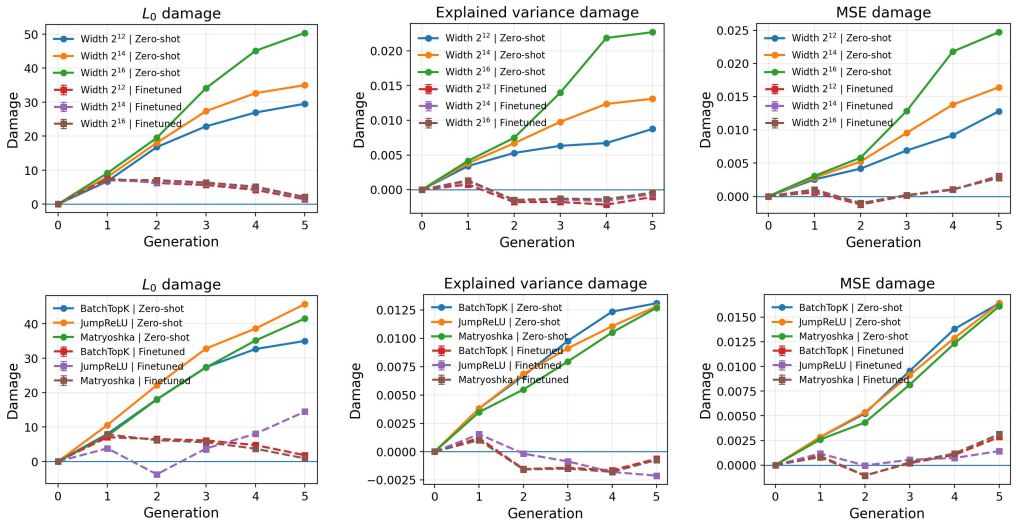

Figure 1: **Zero-shot damage to `Pythia-160M` SAEs grows with self-consumption; fine-tuning removes most of it.** Top: BatchTopK SAEs with widths 2^{12}, 2^{14}, and 2^{16}. Bottom: BatchTopK, JumpReLU, and Matryoshka-BatchTopK SAEs at width 2^{14}. Solid lines denote zero-shot evaluation with the original SAE held fixed; dashed lines denote SAEs fine-tuned separately on each generation (error bars show variation across fine-tuning seeds). All quantities are damage relative to the matched real-data lineage; higher values indicate greater degradation under self-consumption.

Evaluation metrics. We use four SAEBench (Karvonen et al., 2025) evaluation families to separate reconstruction quality from feature-level usefulness (Appendix A.2). *Core metrics* measure sparsity (L_0), reconstruction MSE, and explained variance. We complement these with *sparse probing*, which measures how well task-relevant concepts can be recovered from a small number of SAE latents (Gurnee et al., 2023; Kantamneni et al., 2025). We train probes using the top $k \in \{1, 2, 5\}$ latents and the full SAE representation. Comparing top-k and full-SAE probing distinguishes loss of information from loss of sparse localization: if top-k performance degrades while full-SAE performance remains stable, the information remains present but is distributed across more latents.

Finally, we use two *causal ablation* benchmarks: spurious correlation removal (SCR) and targeted probe perturbation (TPP). We treat sparse probing as our primary measure of feature localization, with SCR and TPP providing complementary evidence about causal feature usefulness.

6 Results

Unless otherwise stated, we report results as damage relative to the generation-matched real-data control (Eq. 3), so that positive values indicate worse performance under self-consumption.

Self-Consumption Progressively Degrades the Underlying Models. We first establish that the self-consuming lineages exhibit cumulative behavioral degradation. As shown in Appendix A.3, Fig. 4, perplexity on held-out real text increases across generations for both synthetic lineages. By generation 5, perplexity is $2.13\times$ that of the matched control for `Pythia-160M` and $1.46\times$ for `Gemma-2-2B`, corresponding to log-perplexity gaps of 0.75 and 0.38 nats, respectively. By comparison, continued training on real data produces much smaller changes in perplexity. Pythia's real-data lineage rises from a perplexity of 23.2 to roughly 30 after the first few generations and then plateaus, while Gemma's control improves with continued real-data training. Thus, the widening synthetic–real gap cannot be explained by continued training alone: recursive training on model-generated data introduces additional degradation that accumulates across generations. Having established this behavioral divergence, we next ask whether self-consumption produces a corresponding shift in the internal representations captured by SAEs.

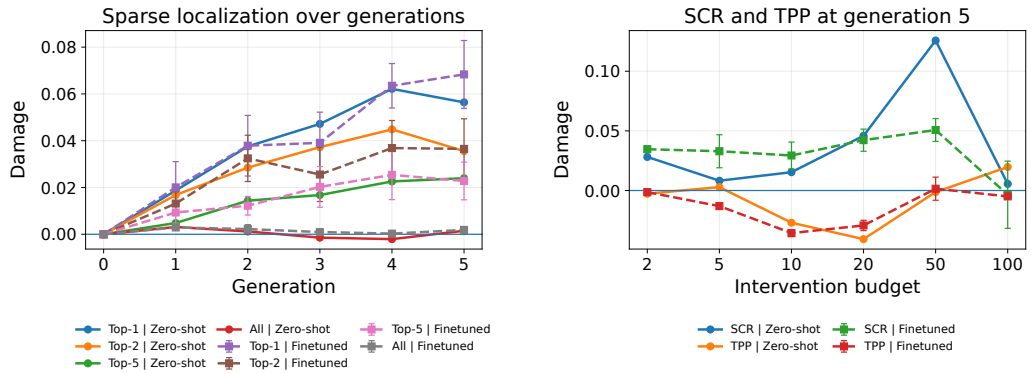

Figure 2: **Fine-tuning does not restore sparse localization in `Pythia-160M`.** Left: sparse-probing damage using the top-k latents or the full SAE representation. Damage remains high for small k after fine-tuning, while full-SAE probing stays near the real-data control, indicating that task-relevant information persists but is less concentrated in individual latents. Right: SCR/TPP damage at gen. 5 across intervention budgets, providing complementary feature-level evaluations.

Pythia: Zero-Shot SAEs Become Progressively Stale. The solid lines in Fig. 1 show core-metric damage when the generation-0 Pythia SAE is applied zero-shot to later checkpoints. Damage increases steadily across generations in L_0, explained variance, and MSE for every configuration. Across widths, the widest dictionary exhibits the largest degradation; across architectures, BatchTopK, JumpReLU, and Matryoshka-BatchTopK follow qualitatively similar trends, indicating that the effect is not specific to a particular SAE design. Because each synthetic checkpoint is compared against the real-data control at the same generation, these trends isolate the additional representational shift induced by self-consumption. This progressive degradation indicates that the original SAE becomes increasingly misaligned to the activation distribution induced by self-consumption.

Pythia: Fine-tuning Restores Reconstruction but Not Sparse Localization. The dashed lines in Fig. 1 show what happens when the original SAE is adapted to each generation. Fine-tuning removes most of the synthetic–real gap in explained variance and MSE and substantially reduces the L_0 gap. Residual sparsity damage remains, most notably for JumpReLU, but is much smaller than under zero-shot transfer. This recovery indicates that the shifted activations remain well captured by a sparse dictionary once the SAE is allowed to adapt, and that much of the zero-shot degradation arises from mismatch between the original dictionary and the evolving representation. Reconstruction recovery, however, does not establish that task-relevant information remains localized in the same way across individual SAE latents.

Feature-level behavior follows a different pattern (Fig. 2, left). Sparse-probing damage increases across generations when probes are restricted to the top few latents, with the largest effect for top-1 probing, and fine-tuning does not eliminate this degradation. In contrast, probes over the full SAE representation show little to no damage in either regime. This divergence suggests that task-relevant information is not lost from the SAE representation, but becomes less concentrated in a small number of individual latents. Self-consumption therefore reduces the sparse localization of task-relevant information even after reconstruction quality has largely recovered. The intervention-based metrics show a less uniform pattern (Fig. 2, right). At generation 5, SCR damage is positive across most intervention budgets and is not consistently removed by fine-tuning, whereas TPP damage remains near zero or changes sign. Thus, feature-level degradation is evident in sparse localization and some causal metrics, but does not appear uniformly across all measures of SAE feature quality.

Gemma: Representational Drift Is Strongly Depth-Dependent. Fig. 3 (top) shows that zero-shot SAE degradation in `Gemma-2-2B` depends strongly on depth. The early layer remains largely stable across generations, the middle layer develops a moderate synthetic–real gap, and the late layer shows large, progressively growing damage in L_0, explained variance, and MSE. Fine-tuning, however, nearly eliminates the explained-variance and MSE gaps at all three layers, including the late

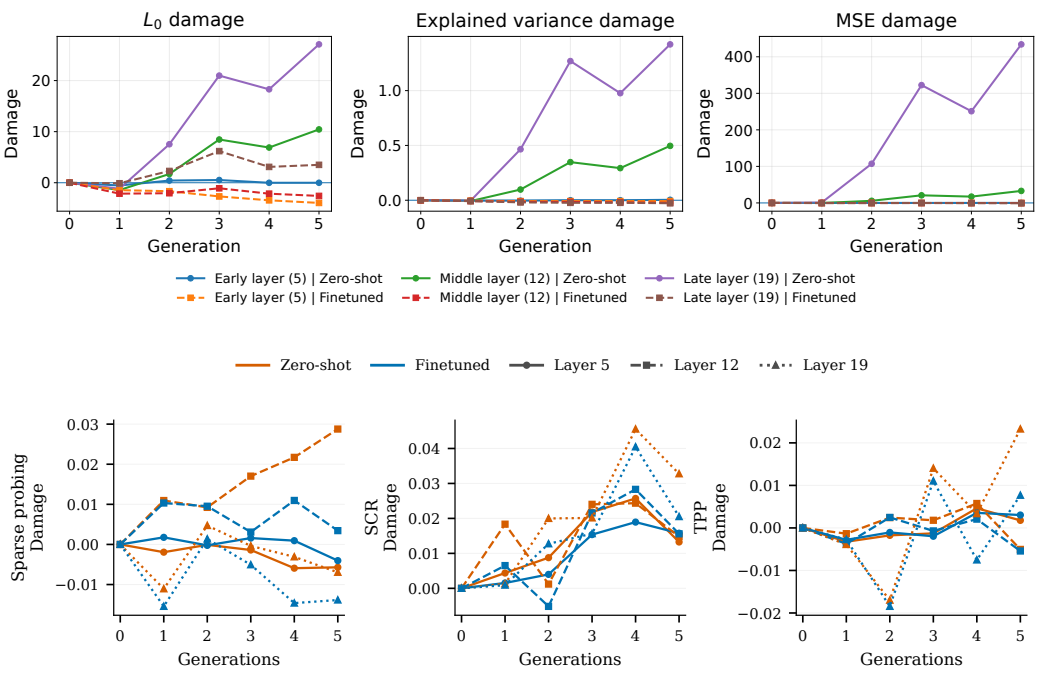

Figure 3: **Self-consumption damage in `Gemma-2-2B` SAEs at an early (5), middle (12), and late (19) layer.** Top: core metrics (solid: zero-shot; dashed: fine-tuned). Zero-shot damage grows with depth and generation; fine-tuning removes nearly all of it, leaving a small residual L_0 gap at layer 19. Bottom: sparse-probing, SCR, and TPP damage; color denotes the regime (orange: zero-shot; blue: fine-tuned) and line style the layer. Higher is worse.

layer, while leaving only a small residual sparsity gap. Thus, self-consumption produces substantially greater representational drift at later layers, but these shifted representations remain aligned to sparse reconstruction once the SAE is adapted.

Gemma: Feature-Level Degradation Is Not Consistent Across Metrics. Gemma's feature-level behavior is less uniform than its reconstruction-level drift (Fig. 3, bottom). For sparse probing, the clearest sustained zero-shot degradation appears at the middle layer, where damage grows over later generations but is substantially reduced by fine-tuning. The early layer remains largely stable, while the late layer fluctuates without a consistent trend. Unlike Pythia, Gemma therefore shows no robust loss of sparse localization after SAE adaptation. The intervention-based metrics are similarly mixed. SCR damage emerges in later generations and is generally larger at deeper layers; fine-tuning reduces this damage but does not consistently eliminate it. TPP shows no clear pattern, with damage changing sign across generations and layers and adaptation sometimes helping and sometimes hurting. Overall, the strong depth-dependent drift seen in reconstruction does not translate into equally strong or persistent degradation across feature-level interpretability metrics.

7 Discussion and Limitations

Self-consumption makes fixed SAE dictionaries stale. Across both model families, SAEs trained before self-consumption transfer progressively worse to later generations. Because each synthetic checkpoint is compared with a generation-matched real-data control, this degradation isolates the effect of synthetic rather than continued training itself. Yet fine-tuning the original SAE removes most of the reconstruction gap, even where zero-shot failure is largest. The representations therefore remain amenable to sparse reconstruction; the main failure is that the original dictionary becomes misaligned with the shifted activation distribution. In practice, an SAE that degrades after model updates may need adaptation rather than replacement.

Representational drift grows with depth. In Gemma, zero-shot degradation is small at an early layer and much larger deeper in the network. This is consistent with prior analyses of Gemma Scope SAEs: early-layer features often track token-level structure, while deeper features capture more conceptual and context-dependent information (Sun et al., 2025). Structural properties may be relatively stable across human- and model-generated text, whereas deeper semantic representations may be more sensitive to the distributional narrowing induced by self-consumption. Our experiments establish the depth dependence but do not identify its mechanism. Importantly, the largest drift appears in the deeper layers that are often most interesting for interpretability.

Reconstruction recovery does not guarantee feature recovery. Pythia provides the clearest example. After fine-tuning, reconstruction largely recovers, but sparse probing with the top few latents remains degraded while probing the full SAE representation does not. Task-relevant information therefore remains accessible but becomes less concentrated in individual features. Gemma shows a more heterogeneous pattern: sparse probing is largely repaired, SCR retains residual degradation, and TPP shows no consistent direction. These metrics capture different properties: localization versus causal usefulness, and need not move together. More broadly, reconstruction metrics can reveal when an SAE has become stale, but successful reconstruction after adaptation is not sufficient to establish that its features remain equally useful for interpretation.

Limitations. Our experiments are intentionally controlled rather than exhaustive. We study two model families over five generations to test whether the same pattern appears across different scales and architectures while keeping the recursive training process tractable. We use pure synthetic training, without mixing in real data, because this isolates the effect of self-consumption; real–synthetic mixtures are more realistic, but would make it harder to attribute changes specifically to synthetic data. We analyze one Pythia layer and early, middle, and late Gemma layers rather than every layer, which lets us test both robustness across SAE designs and variation with depth without greatly expanding the experiment. We also fine-tune the original SAE rather than train a new SAE from scratch because our main question is whether an existing interpretability tool remains reliable as the model evolves. Likewise, we use standardized SAEBench metrics so that reconstruction, localization, and causal behavior can be compared consistently across generations, although these metrics do not capture every aspect of semantic interpretability. Finally, each model lineage uses one training seed because independently training multiple five-generation lineages is computationally expensive; our reported error bars therefore capture variation in SAE fine-tuning rather than variation across model-training runs. Larger models, additional lineage seeds, broader layer coverage, and more realistic mixtures of real and synthetic data are natural next steps.

8 Conclusion

Through this work, we studied how self-consumption changes the internal representations exposed by sparse autoencoders. Across both model families, fixed SAEs become progressively less reliable under self-consumption, with the strongest degradation appearing in later layers. Yet fine-tuning largely restores reconstruction, showing that the representations remain compressible even as the original SAE dictionary becomes stale. The key finding is that reconstruction recovery does not imply feature recovery. In Pythia, task-relevant information remains accessible in the full SAE representation while becoming less concentrated in individual latents; in Gemma, feature-level effects depend on layer and metric. These results highlight an important distinction for interpretability on evolving models: representation shift, SAE adaptation, and feature-level usefulness are separate questions. Reconstruction can tell us whether an SAE still fits the activations, but not whether its features remain equally useful for interpretation.

AI use statement

In this work, we used generative AI tools to assist with writing, code generation, and verification. For writing, we provided the models with raw notes, arguments, and unfiltered draft content and used them to improve clarity, organization, and presentation, including refining portions of the introduction and polishing the manuscript. For visualization, we provided experimental results in

CSV format together with the intended comparisons and plot designs, and used generative AI to produce and iteratively refine plotting code and figure formatting for the ICLR style. We also used generative AI as an additional check of the theoretical analysis, including reviewing derivations for mathematical consistency and identifying potential issues.

We did not use generative AI to generate experimental results or fabricate data. All AI-assisted text was reviewed and edited by the authors; plots were checked against the underlying experimental values; and the theoretical arguments and derivations were independently reviewed for correctness by the authors. We take responsibility for the final content of this work, including text, claims, code, figures, and other artifacts produced with the aid of generative AI.

REFERENCES

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016.

Sina Alemohammad, Josue Casco-Rodriguez, Lorenzo Luzi, Ahmed Imtiaz Humayun, Hossein Babaei, Daniel LeJeune, Ali Siahkoohi, and Richard G. Baraniuk. Self-consuming generative models go MAD. In *International Conference on Learning Representations (ICLR)*, 2024.

Pierre Baldi and Kurt Hornik. Neural networks and principal component analysis: Learning from examples without local minima. *Neural Networks*, 2(1):53–58, 1989.

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

Quentin Bertrand, Avishek Joey Bose, Alexandre Duplessis, Marco Jiralerspong, and Gauthier Gidel. On the stability of iterative retraining of generative models on their own data. In *International Conference on Learning Representations (ICLR)*, 2024.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning (ICML)*, pp. 2397–2430, 2023.

Steven Bills, Nick Cammarata, Dan Mossing, Henk Tillman, Leo Gao, Gabriel Goh, Ilya Sutskever, Jan Leike, Jeff Wu, and William Saunders. Language models can explain neurons in language models. OpenAI, 2023. `https://openaipublic.blob.core.windows.net/neuron-explainer/paper/index.html`.

Hervé Bourlard and Yves Kamp. Auto-association by multilayer perceptrons and singular value decomposition. *Biological Cybernetics*, 59(4):291–294, 1988.

Dan Braun, Jordan Taylor, Nicholas Goldowsky-Dill, and Lee Sharkey. Identifying functionally important features with end-to-end sparse dictionary learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, et al. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023. `https://transformer-circuits.pub/2023/monosemantic-features`.

Martin Briesch, Dominik Sobania, and Franz Rothlauf. Large language models suffer from their own output: An analysis of the self-consuming training loop. *arXiv preprint arXiv:2311.16822*, 2023.

Bart Bussmann, Patrick Leask, and Neel Nanda. BatchTopK sparse autoencoders. *arXiv preprint arXiv:2412.06410*, 2024.

Bart Bussmann, Noa Nabeshima, Adam Karvonen, and Neel Nanda. Learning multi-level features with Matryoshka sparse autoencoders. *arXiv preprint arXiv:2503.17547*, 2025.

David Chanin, James Wilken-Smith, Tomáš Dulka, Hardik Bhatnagar, and Joseph Bloom. A is for absorption: Studying feature splitting and absorption in sparse autoencoders. *arXiv preprint arXiv:2409.14507*, 2024.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. In *International Conference on Learning Representations (ICLR)*, 2024.

Chandler Davis and William M. Kahan. The rotation of eigenvectors by a perturbation. III. *SIAM Journal on Numerical Analysis*, 7(1):1–46, 1970.

Elvis Dohmatob, Yunzhen Feng, Pu Yang, Francois Charton, and Julia Kempe. A tale of tails: Model collapse as a change of scaling laws. In *International Conference on Machine Learning (ICML)*, 2024.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, et al. Toy models of superposition. *Transformer Circuits Thread*, 2022. https://transformer-circuits.pub/2022/toy_model/index.html.

Joshua Engels, Logan Riggs, and Max Tegmark. Decomposing the dark matter of sparse autoencoders. *arXiv preprint arXiv:2410.14670*, 2024.

Yunzhen Feng, Elvis Dohmatob, Pu Yang, Francois Charton, and Julia Kempe. Beyond model collapse: Scaling up with synthesized data requires verification. *arXiv preprint arXiv:2406.07515*, 2024.

Leo Gao, Tom Dupré la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. Scaling and evaluating sparse autoencoders. In *International Conference on Learning Representations (ICLR)*, 2025.

Gemma Team. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.

Matthias Gerstgrasser, Rylan Schaeffer, Apratim Dey, Rafael Rafailov, Henry Sleight, John Hughes, Tomasz Korbak, Rajashree Agrawal, Dhruv Pai, Andrey Gromov, Daniel A. Roberts, Diyi Yang, David L. Donoho, and Sanmi Koyejo. Is model collapse inevitable? breaking the curse of recursion by accumulating real and synthetic data. *arXiv preprint arXiv:2404.01413*, 2024.

Nate Gillman, Michael Freeman, Daksh Aggarwal, Chia-Hong Hsu, Calvin Luo, Yonglong Tian, and Chen Sun. Self-correcting self-consuming loops for generative model training. In *International Conference on Machine Learning (ICML)*, 2024.

Yanzhu Guo, Guokan Shang, Michalis Vazirgiannis, and Chloé Clavel. The curious decline of linguistic diversity: Training language models on synthetic text. In *Findings of the Association for Computational Linguistics: NAACL 2024*, 2024.

Suchit Gupte, Vishnu Kabir Chhabra, and Mohammad Mahdi Khalili. On the transferability of sparse autoencoders for interpreting compressed models. *arXiv preprint arXiv:2507.15977*, 2025.

Suchit Gupte, Xueru Zhang, and Mohammad Mahdi Khalili. When pruning meets interpretability: Preserving sparse autoencoder robustness in llms, 2026. URL https://arxiv.org/abs/2608.25941.

Wes Gurnee, Neel Nanda, Matthew Pauly, Katherine Harvey, Dmitrii Troitskii, and Dimitris Bertsimas. Finding neurons in a haystack: Case studies with sparse probing. *Transactions on Machine Learning Research*, 2023.

Ryuichiro Hataya, Han Bao, and Hiromi Arai. Will large-scale generative models corrupt future datasets? In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 20555–20565, 2023.

Thomas Heap, Tim Lawson, Lucy Farnik, and Laurence Aitchison. Sparse autoencoders can interpret randomly initialized transformers. *arXiv preprint arXiv:2501.17727*, 2025.

Samyak Jain, Robert Kirk, Ekdeep Singh Lubana, Robert P. Dick, Hidenori Tanaka, Edward Grefenstette, Tim Rocktäschel, and David Scott Krueger. Mechanistically analyzing the effects of fine-tuning on procedurally defined tasks. In *International Conference on Learning Representations (ICLR)*, 2024.

Subhash Kantamneni, Joshua Engels, Senthooran Rajamanoharan, Max Tegmark, and Neel Nanda. Are sparse autoencoders useful? a case study in sparse probing. *arXiv preprint arXiv:2502.16681*, 2025.

Adam Karvonen, Can Rager, Johnny Lin, Curt Tigges, Joseph Bloom, David Chanin, Yeu-Tong Lau, Eoin Farrell, Callum McDougall, Kola Ayonrinde, Demian Till, Matthew Wearden, Arthur Conmy, Samuel Marks, and Neel Nanda. SAEBench: A comprehensive benchmark for sparse autoencoders in language model interpretability. In *International Conference on Machine Learning (ICML)*, pp. 29223–29264, 2025.

Joshua Kazdan, Rylan Schaeffer, Apratim Dey, Matthias Gerstgrasser, Rafael Rafailov, David L. Donoho, and Sanmi Koyejo. Collapse or thrive? perils and promises of synthetic data in a self-generating world. *arXiv preprint arXiv:2410.16713*, 2024.

Connor Kissane, Robert Krzyzanowski, Arthur Conmy, and Neel Nanda. SAEs (usually) transfer between base and chat models. AI Alignment Forum, 2024.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International Conference on Machine Learning (ICML)*, pp. 3519–3529, 2019.

Daniel Kunin, Jonathan Bloom, Aleksandrina Goeva, and Cotton Seed. Loss landscapes of regularized linear autoencoders. In *International Conference on Machine Learning (ICML)*, pp. 3560–3569, 2019.

Michael Lan, Philip Torr, Austin Meek, Ashkan Khakzar, David Krueger, and Fazl Barez. Sparse autoencoders reveal universal feature spaces across large language models. *arXiv preprint arXiv:2410.06981*, 2024.

Weixin Liang, Yaohui Zhang, Zhengxuan Wu, Haley Lepp, Wenlong Ji, Xuandong Zhao, Hancheng Cao, Sheng Liu, Siyu He, Zhi Huang, Diyi Yang, Christopher Potts, Christopher D. Manning, and James Y. Zou. Mapping the increasing use of LLMs in scientific papers. *arXiv preprint arXiv:2404.01268*, 2024.

Tom Lieberum, Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Nicolas Sonnerat, Vikrant Varma, János Kramár, Anca Dragan, Rohin Shah, and Neel Nanda. Gemma scope: Open sparse autoencoders everywhere all at once on Gemma 2. *arXiv preprint arXiv:2408.05147*, 2024.

Jack Lindsey, Adly Templeton, Jonathan Marcus, Thomas Conerly, Joshua Batson, and Christopher Olah. Sparse crosscoders for cross-layer features and model diffing. *Transformer Circuits Thread*, 2024. `https://transformer-circuits.pub/2024/crosscoders/index.html`.

Aleksandar Makelov, George Lange, and Neel Nanda. Towards principled evaluations of sparse autoencoders for interpretability and control. *arXiv preprint arXiv:2405.08366*, 2024.

Samuel Marks, Can Rager, Eric J. Michaud, Yonatan Belinkov, David Bau, and Aaron Mueller. Sparse feature circuits: Discovering and editing interpretable causal graphs in language models. In *International Conference on Learning Representations (ICLR)*, 2025.

Gonzalo Martínez, Lauren Watson, Pedro Reviriego, José Alberto Hernández, Marc Juarez, and Rik Sarkar. Combining generative artificial intelligence (AI) and the internet: Heading towards evolution or degradation? *arXiv preprint arXiv:2303.01255*, 2023.

Julian Minder, Clément Dumas, Caden Juang, Bilal Chughtai, and Neel Nanda. Overcoming sparsity artifacts in crosscoders to interpret chat-tuning. *arXiv preprint arXiv:2504.02922*, 2025.

Bruno A. Olshausen and David J. Field. Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision Research*, 37(23):3311–3325, 1997.

Gonçalo Paulo and Nora Belrose. Sparse autoencoders trained on the same data learn different features. *arXiv preprint arXiv:2501.16615*, 2025.

Gonçalo Paulo, Alex Mallen, Caden Juang, and Nora Belrose. Automatically interpreting millions of features in large language models. *arXiv preprint arXiv:2410.13928*, 2024.

Nikhil Prakash, Tamar Rott Shaham, Tal Haklay, Yonatan Belinkov, and David Bau. Fine-tuning enhances existing mechanisms: A case study on entity tracking. In *International Conference on Learning Representations (ICLR)*, 2024.

Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. SVCCA: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Tom Lieberum, Vikrant Varma, János Kramár, Rohin Shah, and Neel Nanda. Improving dictionary learning with gated sparse autoencoders. *arXiv preprint arXiv:2404.16014*, 2024a.

Senthooran Rajamanoharan, Tom Lieberum, Nicolas Sonnerat, Arthur Conmy, Vikrant Varma, János Kramár, and Neel Nanda. Jumping ahead: Improving reconstruction fidelity with JumpReLU sparse autoencoders. *arXiv preprint arXiv:2407.14435*, 2024b.

Mohamed El Amine Seddik, Suei-Wen Chen, Soufiane Hayou, Pierre Youssef, and Merouane Debbah. How bad is training on synthetic data? a statistical analysis of language model collapse. *arXiv preprint arXiv:2404.05090*, 2024.

Lee Sharkey, Bilal Chughtai, Joshua Batson, Jack Lindsey, Jeff Wu, Lucius Bushnaq, Nicholas Goldowsky-Dill, Stefan Heimersheim, Alejandro Ortega, Joseph Bloom, et al. Open problems in mechanistic interpretability. *arXiv preprint arXiv:2501.16496*, 2025.

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Yarin Gal, Nicolas Papernot, and Ross Anderson. The curse of recursion: Training on generated data makes models forget. *arXiv preprint arXiv:2305.17493*, 2023.

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Nicolas Papernot, Ross Anderson, and Yarin Gal. AI models collapse when trained on recursively generated data. *Nature*, 631(8022):755–759, 2024.

Xiaoqing Sun, Alessandro Stolfo, Joshua Engels, Ben Wu, Senthooran Rajamanoharan, Mrinmaya Sachan, and Max Tegmark. Dense SAE latents are features, not bugs. *arXiv preprint arXiv:2506.15679*, 2025.

Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, et al. Scaling monosemanticity: Extracting interpretable features from Claude 3 Sonnet. *Transformer Circuits Thread*, 2024. `https://transformer-circuits.pub/2024/scaling-monosemanticity/`.

Roman Vershynin. *High-Dimensional Probability: An Introduction with Applications in Data Science*. Cambridge University Press, 2018.

Pablo Villalobos, Anson Ho, Jaime Sevilla, Tamay Besiroglu, Lennart Heim, and Marius Hobbhahn. Position: Will we run out of data? limits of LLM scaling based on human-generated data. In *International Conference on Machine Learning (ICML)*, 2024.

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

Table 1: Fine-tuning and sampling settings. Each setting is shared by the real and synthetic lineages of a given model.

	`Pythia-160M`	**Gemma-2-2B**
Checkpoint	`pythia-160m-deduped`	`gemma-2-2b`
Blocks per generation	10,000	10,000
Block length (tokens)	512	512
Tokens per generation	≈5.1M	≈5.1M
Epochs per generation	1	1
Optimizer	AdamW	AdamW
Learning rate	5×10^{-5}	2×10^{-5}
LR schedule	linear, 3% warmup	linear, 3% warmup
Weight decay	0	0
Gradient clipping (max norm)	1.0	1.0
Per-device batch size	16	4
Gradient accumulation steps	2	8
Effective batch size	32	32
Precision	fp32 weights, bf16 autocast	bf16 throughout
Gradient checkpointing	No	Yes
Sampling	$T = 1.0$, top-$p = 1.0$, no top-k	$T = 1.0$, top-$p = 1.0$, no top-k
Held-out evaluation blocks	500	500
Random seed	0	0
SAE layers	8	5, 12, 19
Generations analyzed	0–5	0–5

APPENDIX

A EXPERIMENTAL DETAILS

A.1 TRAINING AND SAMPLING DETAILS

Real data. Real training data are streamed from `monology/pile-uncopyrighted` and shuffled with a buffer of 10,000 documents. Documents are tokenized, an end-of-sequence token is appended to each, and the resulting token stream is split into 512-token blocks. The first 500 blocks form a fixed held-out evaluation set shared by both lineages. The real-data lineage then consumes consecutive, non-overlapping slices of 10,000 blocks, so no real block is reused across generations or overlaps with the evaluation set. Because Pythia was pretrained on the Pile, these blocks are new to the fine-tuning loop but not necessarily new to the model.

Synthetic data. At each generation of the synthetic lineage, all training data are sampled from the previous-generation checkpoint, with no real data mixed in. Each sample starts from a single beginning-of-sequence token and is generated for up to 512 new tokens with temperature 1.0 and top-$p = 1.0$. We explicitly disable top-k sampling so that default generation configurations cannot truncate the tail of the distribution. The prompt token is discarded, each sample is truncated after its first end-of-sequence token, and the samples are concatenated and split into 512-token blocks, mirroring the real-data packing.

Optimization. Each generation is fine-tuned for one epoch on its 10,000 blocks using the settings in Table 1. Settings are identical across the two lineages of each model and differ between models only where noted. Pythia is trained in mixed precision with fp32 master weights. Gemma is trained entirely in bfloat16 with gradient checkpointing. Because bfloat16 master weights can round very small updates to zero, Gemma's effective step size may be smaller than its nominal learning rate, so cross-model comparisons of effect magnitude should be interpreted with caution.

A.2 EVALUATION METRIC DETAILS

All metrics are computed with SAEBench (Karvonen et al., 2025) at the SAE layers listed in Table 1. Following SAEBench, BOS, EOS, and padding tokens are excluded from all evaluations.

Core metrics. Core metrics are computed on 128-token OpenWebText sequences, using 3,200 sequences for the loss-based scores and 32,000 for sparsity statistics. L_0 is the mean number of active latents per token. Reconstruction MSE is the mean squared error between an activation $\boldsymbol{x}$ and its SAE reconstruction $\hat{\boldsymbol{x}}$. Explained variance is the fraction of the variance of $\boldsymbol{x}$ captured by $\hat{\boldsymbol{x}}$.

Sparse probing. Sparse probing uses 35 one-versus-all binary classification tasks drawn from five datasets: profession classification (Bias in Bios), product category and sentiment classification (Amazon Reviews), language identification (Europarl), programming-language identification (GitHub code), and news topic classification (AG News). Each task uses 4,000 training and 1,000 test examples, truncated to 128 tokens. Inputs are passed through the model and encoded by the SAE, and latent activations are mean-pooled over non-padding tokens. The top k latents are selected by the difference in their mean activation between the positive and negative classes, and a logistic-regression probe is trained on them and evaluated on the held-out test examples. We use $k \in \{1, 2, 5\}$ and also train a probe on all SAE latents. The residual-stream baseline $\mathrm{Acc}_{\mathrm{LLM},k}$ applies the same pooling, selection, and probing procedure to the residual-stream dimensions at the same layer, so SPGap_1 compares probes of equal dimension. Reported values are test accuracies averaged over tasks.

Latent selection and ablation for SCR and TPP. Both evaluations select latents by their attribution to a linear probe. The attribution score combines the projection of a latent's decoder direction onto the probe with the difference in the latent's mean activation between inputs with and without the probed concept. Selected latents are zero-ablated, and the original SAE reconstruction error is added back to the modified reconstruction so that the intervention changes only the targeted latents. For both SCR and TPP, we evaluate intervention budgets of $k \in \{2, 5, 10, 20, 50, 100\}$ ablated latents.

Spurious correlation removal (SCR). SCR adapts the SHIFT procedure of Marks et al. (2025). Each task is built from a dataset with an intended label and a spurious label: profession and gender in Bias in Bios, and product category and sentiment in Amazon Reviews. The data are split into a biased subset, in which the two labels are perfectly correlated (for example, only male professors and female nurses), and a balanced subset containing all four combinations. A linear probe trained on the biased subset relies on both signals. We zero-ablate the k latents with the highest attribution to a probe trained on the spurious label and measure the biased probe's accuracy on the intended label over the balanced subset. The score is

$$S_{\mathrm{SCR}} = \frac{A_{\mathrm{abl}} - A_{\mathrm{base}}}{A_{\mathrm{oracle}} - A_{\mathrm{base}}}, \tag{12}$$

where A_{base} and A_{abl} are the biased probe's accuracies before and after ablation, and A_{oracle} is the accuracy of a probe trained directly on the intended label. The biased probe is not retrained after ablation. A score of 1 means the ablation recovers all of the accuracy lost to the spurious correlation, and 0 means it recovers none.

Targeted probe perturbation (TPP). TPP extends this idea to multiclass datasets without requiring correlated labels. For each class c_j, we train a one-versus-rest linear probe with accuracy A_j. For each class c_i, we select the k latents with the highest signed attribution to that class's probe, and let $A_{i,j}$ denote the accuracy of the probe for c_j after the latents selected for c_i are zero-ablated. The score is

$$S_{\mathrm{TPP}} = \operatorname*{mean}_{i=j} \left(A_j - A_{i,j}\right) - \operatorname*{mean}_{i \neq j} \left(A_j - A_{i,j}\right), \tag{13}$$

that is, the mean accuracy drop of each targeted probe minus the mean drop of the non-targeted probes. A high score means that the latents selected for a class damage that class's probe while leaving the other probes largely intact.

A.3 Additional Language-Modeling Results

As a sanity check that the recursive training procedure produces the expected behavioral degradation, we evaluate each generation on held-out real text. Figure 4 shows that perplexity increases progressively along the synthetic lineage relative to the generation-matched real-data control for both model families. By generation 5, the synthetic–real log-perplexity gap reaches 0.75 nats for

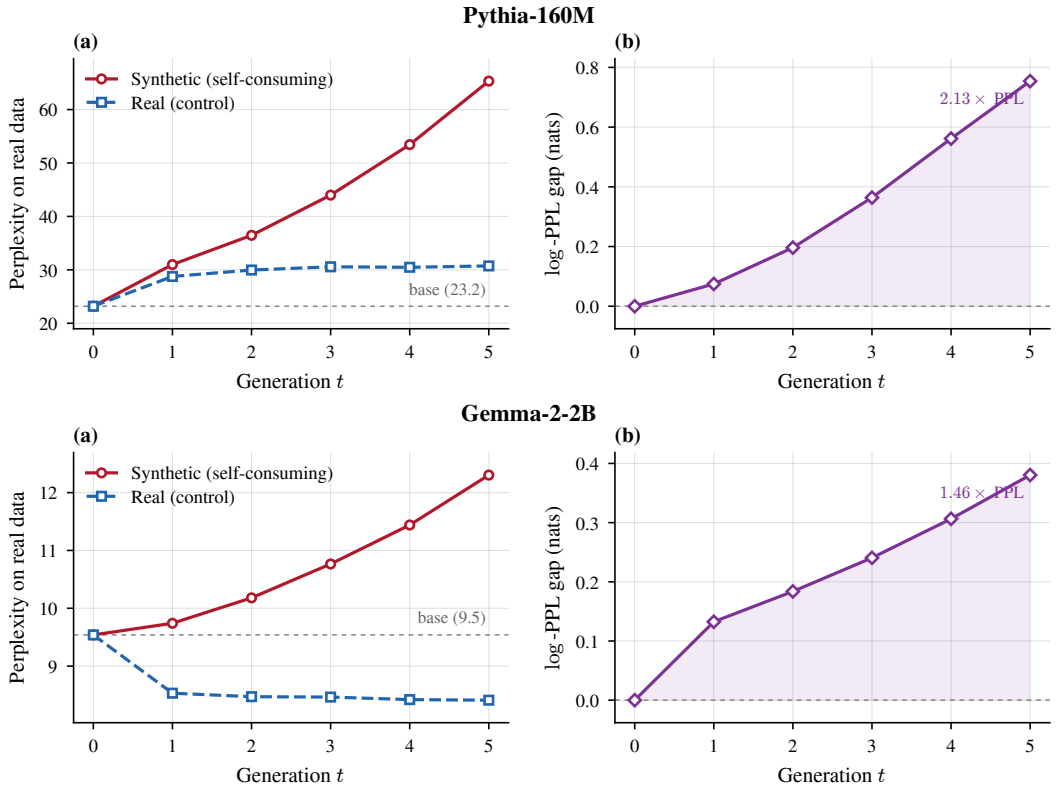

Figure 4: **Self-consumption degrades language modeling.** (a) Perplexity on held-out real data for the synthetic lineage and the matched real-data control. (b) Log-perplexity gap between the two lineages. By generation 5, the gap reaches 0.75 nats (2.13× perplexity) for `Pythia-160M` and 0.38 nats (1.46×) for `Gemma-2-2B`. The control plateaus after the first generation for `Pythia-160M` and improves for `Gemma-2-2B`.

`Pythia-160M` and 0.38 nats for `Gemma-2-2B`, corresponding to 2.13× and 1.46× higher perplexity, respectively. In contrast, continued training on real data produces substantially smaller changes. These results confirm that the self-consuming lineages accumulate degradation beyond that induced by continued training alone, providing the behavioral context for the representational shifts analyzed in the main text.

B Theoretical Details

This appendix provides the derivations underlying the linear-autoencoder analysis in Sec. 4. The analysis is intended as an analytically tractable model of SAE staleness rather than an exact model of nonlinear SAE training. Its main purpose is to separate three quantities: (i) how well the current representation can be reconstructed by an adapted dictionary, (ii) the additional reconstruction cost incurred by keeping an old dictionary fixed, and (iii) the localization of information within individual latent coordinates.

B.1 Linear Autoencoders Recover a Principal Subspace

Let $h \in \mathbb{R}^d$ denote a zero-mean activation vector with covariance

$$\Sigma = \mathbb{E}[hh^\top],$$

and eigendecomposition

$$\Sigma = V \Lambda V^\top, \qquad \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d \geq 0.$$

Consider a rank-k linear autoencoder with decoder $W \in \mathbb{R}^{d\times k}$ satisfying $W^\top W = I_k$ and tied encoder $W^\top$. Its reconstruction is

$$\hat{h} = WW^\top h.$$

Writing $P = WW^\top$, the expected reconstruction loss is

$$\mathcal{L}(P;\Sigma) = \mathbb{E}\|h - Ph\|_2^2 \tag{14}$$
$$= \mathrm{tr}[(I-P)\Sigma] \tag{15}$$
$$= \mathrm{tr}(\Sigma) - \mathrm{tr}(P\Sigma). \tag{16}$$

Proposition B.1 (Linear autoencoder optimum)**.** *Among all rank-k orthogonal projections, the reconstruction loss in Eq. 16 is minimized by projection onto the top-k eigenspace of Σ:*

$$P^\star = V_k V_k^\top,$$

where $V_k = [v_1, \ldots, v_k]$. The corresponding minimum reconstruction error is

$$\mathcal{L}^\star(\Sigma) = \sum_{i=k+1}^{d} \lambda_i.$$

Proof. Since $\mathrm{tr}(\Sigma)$ is independent of P, minimizing Eq. 16 is equivalent to maximizing $\mathrm{tr}(P\Sigma)$. By the Ky Fan maximum principle, the maximizing rank-k projection is the projection onto the eigenvectors corresponding to the k largest eigenvalues. Hence

$$\mathrm{tr}(P^\star\Sigma) = \sum_{i=1}^{k} \lambda_i,$$

and therefore

$$\mathcal{L}^\star = \sum_{i=1}^{d} \lambda_i - \sum_{i=1}^{k} \lambda_i = \sum_{i=k+1}^{d} \lambda_i.$$

□

Thus, in the linear setting, adapting an autoencoder to a new activation distribution corresponds to adapting the subspace used for reconstruction.

B.2 Dictionary Staleness Under Subspace Drift

Let $\Sigma_\ell^{(t)}$ denote the activation covariance at generation t for lineage $\ell \in \{\mathrm{syn}, \mathrm{real}\}$, and let

$$P_\ell^{(t)}$$

denote its optimal rank-k projection. Both lineages share the same generation-0 covariance and therefore the same initial projection

$$P^{(0)} = P_{\mathrm{syn}}^{(0)} = P_{\mathrm{real}}^{(0)}.$$

A frozen generation-0 autoencoder incurs reconstruction error

$$\mathcal{L}_\ell^{\mathrm{zs}}(t) = \mathcal{L}\left(P^{(0)}; \Sigma_\ell^{(t)}\right),$$

whereas an ideally adapted autoencoder obtains

$$\mathcal{L}_\ell^{\mathrm{adapt}}(t) = \mathcal{L}\left(P_\ell^{(t)}; \Sigma_\ell^{(t)}\right).$$

We define the *staleness penalty*

$$S_\ell(t) := \mathcal{L}_\ell^{\mathrm{zs}}(t) - \mathcal{L}_\ell^{\mathrm{adapt}}(t) \tag{17}$$
$$= \mathrm{tr}\left[\left(P_\ell^{(t)} - P^{(0)}\right)\Sigma_\ell^{(t)}\right]. \tag{18}$$

Because $P_\ell^{(t)}$ minimizes reconstruction error for $\Sigma_\ell^{(t)}$,

$$S_\ell(t) \geq 0.$$

The quantity therefore measures exactly how much reconstruction is lost because the dictionary is held fixed rather than adapted to the current representation.

B.2.1 STALENESS IS CONTROLLED BY SUBSPACE MISALIGNMENT

Let the eigenvalues of $\Sigma_\ell^{(t)}$ be

$$\lambda_1^{(t)} \geq \cdots \geq \lambda_d^{(t)},$$

and define the eigengap at the reconstruction boundary as

$$\gamma_\ell^{(t)} = \lambda_k^{(t)} - \lambda_{k+1}^{(t)}.$$

Let

$$\Theta_\ell^{(t)}$$

denote the principal angles between the range of $P^{(0)}$ and the current optimal subspace $P_\ell^{(t)}$.

Proposition B.2 (Staleness and principal angles)**.** *If $\gamma_\ell^{(t)} > 0$, then*

$$\gamma_\ell^{(t)} \left\| \sin \Theta_\ell^{(t)} \right\|_F^2 \leq S_\ell(t) \leq \left(\lambda_1^{(t)} - \lambda_d^{(t)} \right) \left\| \sin \Theta_\ell^{(t)} \right\|_F^2 . \tag{19}$$

Proof. Let $u_1, \ldots, u_d$ denote the eigenvectors of $\Sigma_\ell^{(t)}$ and define

$$a_i = u_i^\top P^{(0)} u_i.$$

Since $P^{(0)}$ is a rank-k projection, $0 \leq a_i \leq 1$ and

$$\sum_{i=1}^{d} a_i = k.$$

The staleness penalty can be written as

$$S_\ell(t) = \sum_{i=1}^{k} \lambda_i^{(t)} - \sum_{i=1}^{d} \lambda_i^{(t)} a_i \tag{20}$$

$$= \sum_{i=1}^{k} \lambda_i^{(t)} (1 - a_i) - \sum_{i=k+1}^{d} \lambda_i^{(t)} a_i. \tag{21}$$

Moreover,

$$s := \sum_{i=1}^{k} (1 - a_i) = \sum_{i=k+1}^{d} a_i = \| \sin \Theta_\ell^{(t)} \|_F^2.$$

Using $\lambda_i^{(t)} \geq \lambda_k^{(t)}$ for $i \leq k$ and $\lambda_i^{(t)} \leq \lambda_{k+1}^{(t)}$ for $i > k$,

$$S_\ell(t) \geq \left(\lambda_k^{(t)} - \lambda_{k+1}^{(t)} \right) s.$$

Similarly,

$$S_\ell(t) \leq \left(\lambda_1^{(t)} - \lambda_d^{(t)} \right) s.$$

Substituting $s = \| \sin \Theta_\ell^{(t)} \|_F^2$ gives Eq. 19. □

Therefore, degradation of a frozen dictionary is directly tied to movement of the reconstruction-relevant subspace. A covariance shift that changes only eigenvalues need not make the dictionary stale; a shift that rotates the principal subspace does.

B.3 Covariance Perturbations and Subspace Drift

Write the covariance at generation t as

$$\Sigma_\ell^{(t)} = \Sigma^{(0)} + \Delta_\ell^{(t)}.$$

Let

$$\gamma^{(0)} = \lambda_k(\Sigma^{(0)}) - \lambda_{k+1}(\Sigma^{(0)})$$

be the generation-0 eigengap.

Standard eigenspace perturbation results imply that, while the perturbation remains small relative to this eigengap,

$$\left\|\sin \Theta_\ell^{(t)}\right\|_F \lesssim \frac{\sqrt{k}\|\Delta_\ell^{(t)}\|_{\mathrm{op}}}{\gamma^{(0)}}. \tag{22}$$

Combining Eqs. 19 and 22 gives

$$S_\ell(t) \lesssim \left(\lambda_1^{(t)} - \lambda_d^{(t)}\right) \frac{k\|\Delta_\ell^{(t)}\|_{\mathrm{op}}^2}{(\gamma^{(0)})^2}. \tag{23}$$

The bound is not intended to predict the exact empirical magnitude of SAE damage. Rather, it establishes the qualitative mechanism needed by the main paper: cumulative covariance shift can move the optimal reconstruction subspace, causing an SAE trained at generation 0 to become increasingly mismatched.

It also highlights the role of spectral structure. For the same covariance perturbation, a representation with a small eigengap can undergo greater subspace movement than one with a well-separated principal subspace.

B.4 Matched-Control Decomposition

The experiments compare every synthetic checkpoint against a real-data checkpoint receiving the same continued-training budget. The same comparison can be made in the linear model.

For reconstruction error, define

$$D_{\mathrm{zs}}(t) = \mathcal{L}_{\mathrm{syn}}^{\mathrm{zs}}(t) - \mathcal{L}_{\mathrm{real}}^{\mathrm{zs}}(t), \tag{24}$$

$$D_{\mathrm{adapt}}(t) = \mathcal{L}_{\mathrm{syn}}^{\mathrm{adapt}}(t) - \mathcal{L}_{\mathrm{real}}^{\mathrm{adapt}}(t). \tag{25}$$

Since

$$\mathcal{L}_\ell^{\mathrm{zs}}(t) = \mathcal{L}_\ell^{\mathrm{adapt}}(t) + S_\ell(t),$$

we obtain the exact decomposition

$$D_{\mathrm{zs}}(t) - D_{\mathrm{adapt}}(t) = S_{\mathrm{syn}}(t) - S_{\mathrm{real}}(t). \tag{26}$$

Equation 26 gives a direct interpretation of the gap between zero-shot and adapted evaluation. If self-consumption causes more dictionary mismatch than matched continued training on real data, then

$$S_{\mathrm{syn}}(t) > S_{\mathrm{real}}(t),$$

and adapting the dictionary reduces the synthetic–real reconstruction gap.

Importantly, a small value of $D_{\mathrm{adapt}}(t)$ does not imply that $\Sigma_{\mathrm{syn}}^{(t)}$ is unchanged. It only means that, after allowing the reconstruction subspace to adapt, the synthetic representation is not substantially harder to reconstruct than the matched real-data representation.

B.5 Reconstruction Does Not Identify Individual Features

The preceding analysis concerns the subspace captured by an autoencoder. Feature-level evaluations, however, depend on how information is arranged *within* that subspace.

This distinction can be seen even in the linear setting. Let $W \in \mathbb{R}^{d\times k}$ be an orthonormal decoder and let $Q \in \mathbb{R}^{k\times k}$ be any orthogonal matrix. Define

$$W' = WQ.$$

Then

$$W'W'^{\top} = WQQ^{\top}W^{\top} = WW^{\top}. \tag{27}$$

Thus W and W' give exactly the same reconstruction for every input.

Their latent coordinates are nevertheless different:

$$z = W^{\top}h, \qquad z' = W'^{\top}h = Q^{\top}z.$$

Proposition B.3 (Full-space probing is invariant to rotations)**.** *For any linear predictor $a^{\top}z$ defined on the original latent representation, there exists a predictor on z' that gives exactly the same output for every input.*

Proof. Choose

$$a' = Q^{\top}a.$$

Then

$$(a')^{\top}z' = a^{\top}QQ^{\top}z = a^{\top}z.$$

□

Thus a probe with access to the complete latent representation can recover the same information after an invertible orthogonal change of coordinates.

Sparse coordinate-based probes do not have this invariance. Consider, for example, a two-dimensional representation where the task-relevant signal is localized entirely in z_1. Under the rotation

$$Q = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & -1\\ 1 & 1\end{bmatrix},$$

the same signal is distributed across both rotated coordinates. Reconstruction is unchanged, and a probe using both coordinates loses no information, but neither individual coordinate need retain the original level of predictive power.

This example establishes a one-way implication:

$$\text{good reconstruction} \not\Rightarrow \text{information localized in individual latents.}$$

It does *not* imply that SAE fine-tuning necessarily rotates features or necessarily reduces sparse localization. Rather, it shows why reconstruction recovery alone cannot establish recovery of feature-level usefulness.

B.6 An Illustrative Recursive Covariance Model

We next revisit a simple recursive Gaussian model to illustrate how repeated training can accumulate covariance perturbations. This model is deliberately stylized and is not intended as a mechanistic description of language-model self-consumption.

Let the generation-0 distribution be

$$h^{(0)} \sim \mathcal{N}(0, \Sigma^{(0)}).$$

Consider first the idealized population update

$$\Sigma_{\star}^{(t+1)} = \Sigma_{\star}^{(t)} + \sigma^2 I, \qquad \Sigma_{\star}^{(0)} = \Sigma^{(0)}. \tag{28}$$

Hence

$$\Sigma_{\star}^{(t)} = \Sigma^{(0)} + t\sigma^2 I. \tag{29}$$

B.6.1 ISOTROPIC POPULATION DRIFT IS NOT SUFFICIENT FOR DICTIONARY STALENESS

If

$$\Sigma^{(0)} = V\Lambda^{(0)}V^\top,$$

then

$$\Sigma_\star^{(t)} = V\left(\Lambda^{(0)} + t\sigma^2 I\right)V^\top.$$

The eigenvectors are therefore identical at every generation, and

$$P_\star^{(t)} = P^{(0)}.$$

Consequently,

$$S_\star(t) = 0.$$

This observation is useful because it identifies what the theory requires: *variance growth alone does not make a linear dictionary stale*. The staleness observed in the experiments requires a shift that changes the reconstruction-relevant geometry, such as anisotropic covariance drift, finite-sample perturbations, or other changes induced by neural-network training.

B.6.2 FINITE-SAMPLE PERTURBATIONS

Suppose that at each generation only n samples are available. Let $\widehat{\Sigma}^{(t)}$ denote the empirical covariance and define

$$E^{(t)} = \widehat{\Sigma}^{(t)} - \Sigma_\star^{(t)}.$$

The actual covariance used to generate the next sample is

$$\widetilde{\Sigma}^{(t)} = \widehat{\Sigma}^{(t-1)} + \sigma^2 I.$$

Conditional on $\widetilde{\Sigma}^{(t)}$, standard concentration for Gaussian sample covariance gives, with high probability,

$$\left\|\widehat{\Sigma}^{(t)} - \widetilde{\Sigma}^{(t)}\right\|_{\mathrm{op}} \le \epsilon_{n,d,\delta}\left\|\widetilde{\Sigma}^{(t)}\right\|_{\mathrm{op}}, \tag{30}$$

where

$$\epsilon_{n,d,\delta} = C\left[\sqrt{\frac{d+\log(1/\delta)}{n}} + \frac{d+\log(1/\delta)}{n}\right]$$

for an absolute constant C.

Because

$$\widetilde{\Sigma}^{(t)} - \Sigma_\star^{(t)} = E^{(t-1)},$$

we have

$$\|E^{(t)}\|_{\mathrm{op}} \le \left\|\widehat{\Sigma}^{(t)} - \widetilde{\Sigma}^{(t)}\right\|_{\mathrm{op}} + \|E^{(t-1)}\|_{\mathrm{op}} \tag{31}$$

$$\le (1+\epsilon)\|E^{(t-1)}\|_{\mathrm{op}} + \epsilon\left(\lambda_1^{(0)} + t\sigma^2\right), \tag{32}$$

where $\epsilon = \epsilon_{n,d,\delta/T}$ after a union bound over generations.

If $E^{(0)} = 0$, unrolling the recursion gives

$$\|E^{(t)}\|_{\mathrm{op}} \le \epsilon\sum_{s=1}^{t}(1+\epsilon)^{t-s}\left(\lambda_1^{(0)} + s\sigma^2\right). \tag{33}$$

Unlike the isotropic population term, $E^{(t)}$ is generally anisotropic and can therefore rotate empirical principal directions. Applying Eq. 22 to this perturbation shows how recursive finite-sample error can create an increasingly stale empirical dictionary.

We emphasize that this construction is an illustrative example of the more general covariance-drift mechanism in Sec. 4. In the language models studied experimentally, representational changes can arise from many sources beyond finite-sample covariance estimation.

B.7 Threshold-Based Sparsity Under Variance Growth

The isotropic population model can nevertheless illustrate how activation scale alone affects a threshold-based notion of sparsity.

For the ith principal component,

$$z_i^{(t)} = v_i^\top h^{(t)},$$

and under Eq. 29,

$$z_i^{(t)} \sim \mathcal{N}\left(0, \lambda_i^{(0)} + t\sigma^2\right).$$

For a fixed firing threshold $\tau > 0$,

$$p_i^{(t)} := \Pr\left(|z_i^{(t)}| > \tau\right) \tag{34}$$

$$= 2\Phi\left(-\frac{\tau}{\sqrt{\lambda_i^{(0)} + t\sigma^2}}\right), \tag{35}$$

where Φ is the standard normal CDF.

Since the variance increases with t, $p_i^{(t)}$ increases monotonically. Equivalently, the fraction of inactive coordinates

$$\Psi^{(t)} = 1 - \frac{1}{k}\sum_{i=1}^{k} p_i^{(t)} \tag{36}$$

decreases with generation. This calculation provides a simple mechanism by which increasing activation variance can increase a threshold-based L_0 quantity even without a rotation of the underlying subspace. It should therefore be interpreted separately from the staleness analysis above. In particular, because isotropic variance growth leaves the optimal linear dictionary unchanged, this model alone cannot explain why SAE fine-tuning reduces the L_0 gap in the experiments.

B.8 Scope of the Theory

The analysis above is intentionally limited. First, a linear autoencoder is a proxy for an SAE, not an exact model of a nonlinear sparse dictionary. The PCA equivalence is used to obtain a tractable notion of reconstruction-relevant representational drift. Second, the theory does not predict that self-consumption must produce monotonic degradation in every SAE metric. It predicts that movement of the reconstruction-relevant subspace creates a penalty for a frozen dictionary and that adapting the dictionary can remove this penalty. Third, reconstruction recovery places no guarantee on the organization of individual features. Consequently, the theory is compatible both with the persistent sparse-localization degradation observed in Pythia and with the more heterogeneous feature-level effects observed across Gemma layers. Finally, the theory does not explain why particular model layers drift more than others. The depth dependence observed empirically should therefore be viewed as an experimental finding: the analysis states that greater subspace drift produces greater dictionary staleness, but does not predict which layers should undergo that drift.