

Written Is Not Kept: What a Single-Layer Readout of the Residual Stream Licenses When Adjacent Layers Undo Part of Each Other's Writes, Why Cancellation and Iterative Refinement Are Not Rival Findings, and the Persistence Check Interpretability Claims Skip

Pranay Mahendrakar

SONYTECH · ORCID 0009-0003-7224-029X

pranaymahendrakar@sonytech.in · mahendrakarpranay@gmail.com · +91 6361723454

Abstract. Most feature-based interpretability reads a transformer one layer at a time. A probe, a lens or a sparse dictionary is fitted to the residual stream, or to one layer's output, at a chosen depth, and what it finds there is reported as something the model has represented and will go on to use. A 2026 study reports that in five of seven open-weight model families the write of each layer points, on average, partly against the write of the layer before it, that the opposition is learned during pretraining, scales with the size of the preceding write and is concentrated in a subspace, and proposes that layers put forward candidate features which the next layer strikes out. Against this stand results that adjacent residual states are almost parallel, that predictions converge smoothly through depth, that dictionary features can be matched across several layers, and that whole middle layers can be removed or swapped at modest cost. In this analysis the author reviews and synthesises published results to show that the two bodies of evidence measure different objects - the write a layer adds, and the state the writes add up to - and that the arithmetic of a residual sum lets writes partly cancel while successive states stay nearly parallel, so neither side refutes the other. What the cancellation evidence does threaten is narrower and more practical: the inference from "present at layer L" to "kept", "used" or "endorsed by the model". The paper separates five claims that a single-layer readout is used to support, distinguishes four published kinds of non-persistence that license different conclusions, shows that the three downstream consequences proposed for layer correction each have a competing explanation the record supports at least as well - including a ten-fold steering coefficient reported for GPT-2-XL, a model in which the correction pattern was not found - consolidates 26 published results in one table and one figure, states a persistence check as an algorithm, and names five studies that would settle the open part.

Keywords: mechanistic interpretability; residual stream; sparse autoencoders; layer correction; logit lens; activation steering; transcoders; iterative inference

SONYTECH ORCID: 0009-0003-7224-029X

1. Introduction

The residual stream of a transformer is a running sum. Each layer reads the current sum, computes something, and adds its result back in. Interpretability methods treat this sum as a place where information can be looked up. A linear probe trained at layer 14 that recovers the part of speech of a token is read as showing that layer 14 represents part of speech. A logit lens or tuned lens applied at layer 20 that ranks the eventual answer first is read as showing that the model has, by layer 20, settled on that answer (Belrose et al., 2023). A sparse autoencoder trained on the activations at one depth returns latents that are labelled, scored and then used as handles for steering (Cunningham et al., 2023; Gao et al., 2024). Behind all of these sits an assumption that is rarely stated because it seems to follow from the architecture: what a layer writes into the sum stays there, so later layers can build on it.

Patrawala, Feng, Jones and Steinhardt (2026) put that assumption under direct test and report that, in most of the open-weight models they examined, it fails in a specific and systematic way. Averaged over about

100,000 tokens of WikiText, the vector that layer $i+1$ adds to the residual stream has a cosine similarity of roughly -0.2 with the vector layer i added, in five of seven model families (Llama 3, OLMo, Mistral, Gemma and Qwen2), while GPT-2 and Phi show no such pattern. Non-adjacent layers do not oppose each other on average. The authors name the effect the Transformer Layer Correction Mechanism and read it through a "propose-and-reject" hypothesis: a layer writes several candidate features and the next layer, having gathered context through attention, removes the inappropriate ones. If that is right, the residual stream at any depth holds both features that will persist and proposals that are about to be withdrawn, and a method that decomposes one layer cannot tell them apart.

The study arrives in a literature that already disagrees about what depth does. One line of work holds that residual networks refine a common estimate step by step (Greff et al., 2016; Jastrzebski et al., 2017), and finds in transformers that hidden states drift smoothly toward the output, that a learned lens can read a sensible next-token distribution from almost every layer (Belrose et al., 2023), that adjacent states are close to linearly related (Razhigaev et al., 2024), and that whole middle layers

can be deleted or swapped with modest loss (Lad et al., 2024; Gromov et al., 2024). Another line reports components whose ordinary job is to push against the signal: late layers that lower the probability of the most likely token (McGrath et al., 2023), a head that suppresses tokens earlier components have predicted (McDougall et al., 2023), neurons that erase or anti-erase (Rushing and Nanda, 2024), units whose response to an intervention follows a fixed negative gain (Ahmad et al., 2026), and early layers that wipe out token-level information once a multi-token word has been assembled (Feucht et al., 2024). Layer correction looks at first like a decisive point for the second line against the first.

This paper argues that it is not, and that the more useful question is a different one. In this analysis the author reviews and synthesises published results to show that cancellation between adjacent writes and convergence of adjacent states are compatible facts about a residual sum, that the evidence for persistence and the evidence against it mostly measure different quantities, and that what is genuinely at stake is a set of inferences interpretability practice makes from a readout at one depth. The contributions are as follows. (1) A separation of five claims - presence, persistence, use, endorsement and origin - that a single-layer readout is routinely taken to support (Section 3). (2) A derivation showing why writes can anti-correlate while states stay nearly parallel, and why "iterative inference" names a claim about states that layer correction does not contradict (Section 4). (3) A classification of four published kinds of non-persistence - rejection, consumption, regulation and re-encoding - with the different conclusions each licenses (Section 5). (4) A reading of the dictionary-based persistence evidence that shows how much of it measures the reuse of directions rather than the survival of a token's feature, and how much of the apparent transience is produced by the dictionary objective (Section 6). (5) A reading of the layer deletion and swapping results as a test of the propose-and-reject hypothesis whose coarse, family-level form is already published and whose pair-level form has not been run (Section 7). (6) A check of the three downstream consequences Patrawala et al. propose against competing explanations in the record (Section 8). (7) Table 1, which consolidates 26 published results, Figure 1, which plots the subset that reports counts or layer spans, and Algorithm 1, a persistence check to run before a single-layer claim is made (Sections 9 and 10). Section 11 states the strongest case against this reading, Section 12 the studies that would settle the open part, and Section 13 what is not known.

Where a value in this paper is derived from published values rather than reported directly, the text marks it "by arithmetic".

2. What the anchor study reports

Patrawala et al. (2026) define the contribution of layer i as everything the layer adds to the residual stream: the attention output plus the MLP output, so that the state

after the layer equals the state before it plus the contribution. For each token they compute the cosine similarity between the contributions of every pair of layers and average the resulting matrix over roughly 100,000 tokens of WikiText, which includes several languages and code. The matrix shows negative bands just off the diagonal in the first two-thirds of the layers. The adjacent-layer average is about -0.2, typical values run from -0.3 to -0.15, and the most negative reaches -0.4. Pairs further apart are on average orthogonal or positively aligned. The study's primary model is Llama 3.1 8B, with a 4096-dimensional residual stream.

Three calibrations matter for what follows. First, the distribution of adjacent-layer cosines across tokens and layers is bimodal, with a main mode near -0.2, a smaller one near +0.1 and a valley near -0.1, and the authors define a "TLCM activation" as any adjacent pair at or below -0.1. Second, they compare the size of the effect to two null models. If two 4096-dimensional vectors were drawn independently from a standard normal distribution, their cosine would have variance $1/4096$, so a value of -0.1 would lie about six standard deviations below zero; if each layer's output were instead drawn independently from that layer's empirical distribution, a typical activation corresponds to about three standard deviations. The authors note that both nulls assume independence that the architecture rules out, since layer $i+1$ reads a sum that includes layer i 's write, and they present the comparison as calibration rather than as evidence of cause. Third, in GPT-2 and in the Phi models the bands are absent; the paper suggests the dropout blocks those models place after their sublayers as one possible reason.

The observational results that follow are these. In three fully open OLMo models the pattern is absent at early checkpoints: at step 33,000 of OLMo 7B, 5 percent of training, the matrix shows no consistent negative structure next to the diagonal; by step 135,500, about 21 percent of training and roughly half a trillion tokens, negative values appear between adjacent middle layers, and they strengthen and spread through the rest of training. The authors conclude that the mechanism is learned during pretraining and not removed by weight decay. On 79,000 tokens of Llama 3.1 8B responses to 100 long-form queries, the count of TLCM activations per token varies by token type: numbers, dates, punctuation, brackets and newlines are corrected most often (the newline token averages 12.41 activations), common content words least often ("sustainability" averages 7.63), and the beginning-of-text token not at all. Over 2,000 long WikiText documents, every 1,000 additional tokens of context adds about one activation per token. At the sublayer level, attention outputs are positively aligned with other attention outputs, while each MLP output is anti-aligned with the previous MLP output and with the preceding attention output, which the authors read as MLPs executing most of the correction with attention contributing.

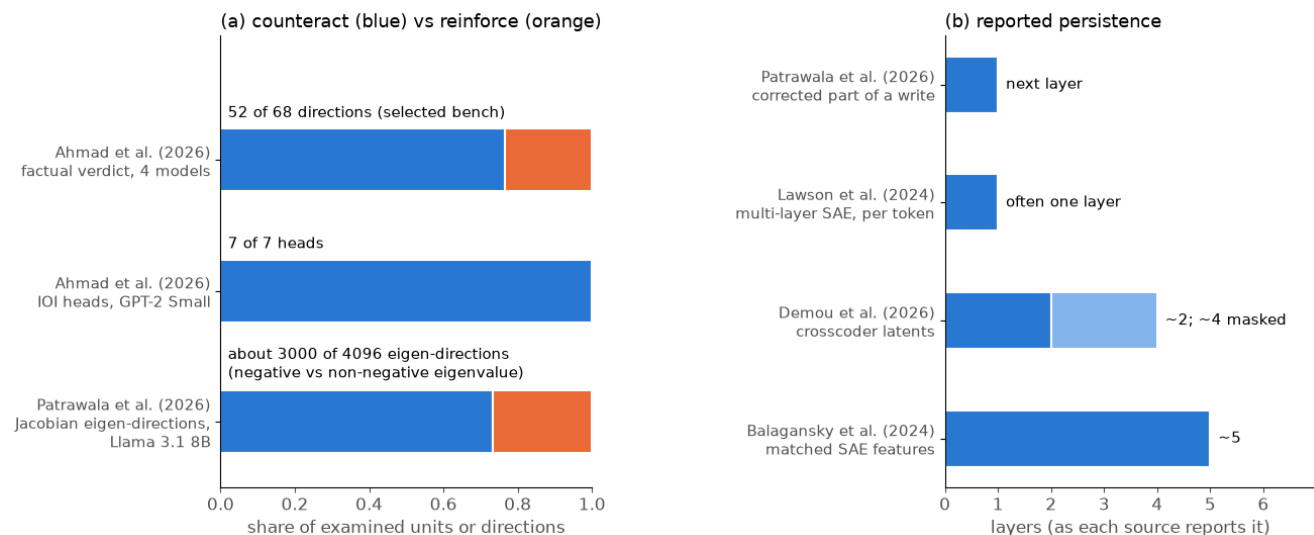


Figure 1. (a) Share of examined units or directions whose effect opposes the incoming signal (blue) or reinforces it (orange): Ahmad et al. (2026), 52 counterweights and 16 relays among 68 certified directions on a factual-verdict task in Gemma, Qwen, Llama and Mistral models, and 7 of 7 law-following heads in the GPT-2 Small indirect-object circuit; Patrawala et al. (2026), eigen-directions of the symmetrised Jacobian for a correcting layer pair in Llama 3.1 8B, about 3000 of 4096 with a negative eigenvalue (blue) and the rest non-negative (orange). (b) Number of layers over which content is reported to last, each in its source's own measure: the corrected part of a write is reversed by the next layer (Patrawala et al., 2026); multi-layer autoencoder latents are often active at a single layer for a given token (Lawson et al., 2024); standard crosscoder latents depend on about 2 layers, about 4 with layer masking (Demou et al., 2026, Pythia-410M); matched autoencoder latents stay significantly similar for about 5 layers (Balagansky et al., 2024, Gemma 2 2B).

The interventional results are the ones that carry the mechanistic claim. Adding a multiple alpha of layer i 's write back into the input of layer $i + 1$, for alpha between -1 and 1, changes the cosine between the two writes almost linearly: more of the earlier write draws more opposition, less draws less, and at alpha = -1 the cosine turns positive, which the authors describe as a compensatory boost and relate to self-repair. Because the response tracks the perturbation, the opposition is not just a shared confounder acting on both layers. Linearising layer $i + 1$ around the clean input and symmetrising its Jacobian, the authors project layer i 's write onto the eigenvectors that carry most of its energy (the top fifty directions account for more than 15 percent of its variance) and look at the eigenvalues. For pairs that show the effect, the eigenvalues cluster near -1, a one-for-one reversal, and near +0.5, a partial reinforcement; for adjacent pairs that do not, they cluster near 0. About 3,000 of the 4,096 eigenvalues of a TLMC Jacobian are negative. The linearisation has about 5 percent error for perturbations with $|\alpha| < 0.1$. An appendix reports that the correction weakens when the earlier write is scaled up by large factors, and another appendix argues that the "LayerNorm blindness" of pre-normalised sublayers does not account for the effect alone: Gemma 2 adds post-normalisation and OLMo 2 replaces pre-normalisation, and both still show it.

The propose-and-reject hypothesis is offered as an interpretation of these results. The authors write that they "do not explicitly test P&R in this work", and the hypothesis has no formal statement. In their discussion they connect it to three open problems: that sparse autoencoder feature descriptions often have low specificity, that steering by a feature direction often

needs very large amplification, and that transcoders produce more interpretable features than sparse autoencoders. For the first two they cite figures from an industry report (Templeton et al., 2024, as cited by Patrawala et al.): that over 50 percent of activated features were judged by a language model to be irrelevant or only vaguely related to the text they fired on, and that effective steering sometimes required up to ten times a feature's maximum observed activation. Section 8 examines each of the three connections.

The study is careful and its measurements are not in question here. What this paper asks is what they license, and what the rest of the literature adds when it is read next to them.

3. Five claims a single-layer readout is asked to carry

When an interpretability paper reports a finding at one layer, the finding is usually some version of "f is at layer L". That phrase is asked to carry at least five different claims, and they come apart.

Presence. A property f is linearly decodable at the site that was read: a probe, a lens or a dictionary latent recovers it from the activation at depth L, above a stated baseline. This is what the readout measures directly.

Persistence. f remains decodable at later depths $L + 1$, ..., $L + k$, so that downstream computation could read it. Persistence has a horizon, and a claim of persistence is a claim about that horizon.

Use. Some downstream computation depends on f : intervening on the direction that carries f at depth L changes what later components do or what the model

outputs, in the direction the label predicts.

Endorsement. The model's eventual behaviour reflects f : the answer a lens ranks first at layer L is the answer the model gives, or the feature a dictionary assigns to a token describes something the model "takes to be true" of that token. Endorsement is what most narrative uses of lens and dictionary results assume - "the model has decided", "the model knows" - and it is the strongest of the five.

Origin. f was written at depth L , not inherited from an earlier layer and carried forward. A readout of the residual state cannot establish origin; a readout of a layer's own contribution can, but only for the part of f that layer added.

A readout supports presence directly. Each of the other four needs a further measurement: persistence needs readouts at later depths, use needs an intervention, endorsement needs a comparison with the output, and origin needs the contribution rather than the state. The anchor study bears on persistence and endorsement, and on origin through its choice of object. It does not bear on presence: a proposal that the next layer withdraws was, by hypothesis, present.

The distinction between the state and the contribution runs through the rest of the paper. The residual state after layer i is the sum of the embedding and every contribution up to i . A sparse autoencoder trained on the residual stream decomposes the state; one trained on an MLP's output decomposes a contribution; a transcoder maps a sublayer's input to its output. Patrawala et al. measure contributions. Most of the evidence for persistence measures states.

4. Contributions versus states: why cancellation and convergence do not conflict

4.1 The arithmetic of a residual sum

Write the state before layer i as x , layer i 's contribution as a and layer $i+1$'s as b , so the next two states are $x + a$ and $x + a + b$. The anchor study measures $\cos(a, b)$. The persistence evidence of the kind discussed in this section measures $\cos(x + a, x + a + b)$, or a closely related similarity between successive states. When the writes are small relative to the stream, the first constrains the second only weakly, as the next two paragraphs show.

The part of b that lies along a has length $|b| \cos(a, b)$. With $\cos(a, b) = -0.2$, and if the two writes have similar lengths, layer $i+1$ removes about one fifth of the length of a along a 's own direction (by arithmetic, under the stated assumption about lengths, which the anchor study does not report). The other four fifths of a , and every component of a in directions b does not touch, survive into the state after layer $i+1$. A cosine of -0.2 between successive writes therefore describes a partial retraction, not an erasure, unless the opposition is concentrated - which is exactly what the Jacobian analysis says it is. If a 's projection onto the corrected subspace is reversed

one-for-one, that part of a is gone after one layer, and the rest of a is not.

The state side follows from the size of the writes relative to the stream. For any vectors u and b , the angle between u and $u + b$ is at most $\arcsin(|b|/|u|)$ when $|b| < |u|$. If each write is a tenth of the stream's norm, successive states have a cosine of at least about 0.995 whatever the sign of $\cos(a, b)$ (by arithmetic, for that illustrative ratio). Published measurements are consistent with this picture, although none reports the ratio in the form used here. Csordas, Manning and Potts (2025) measure the norm of each sublayer's output relative to the residual it is added to across Llama 3.1, Qwen 3 and OLMo 2, report that the residual norm grows rapidly through depth, and find that the cosine similarity of neighbouring residual states is very high, often close to 1, which they describe as "a consequence of the known anisotropy of Transformers". On their reading the high state cosine is largely a property of the representation space, so it is weaker evidence for the small-write arithmetic than it first appears: it is consistent with that arithmetic but does not isolate it. Razzhigaev et al. (2024) report a Procrustes similarity of 0.99 between the embeddings of sequential layers across several decoder families, and add a relevant detail: the linearity falls when the residual component is removed, because each layer's own output has consistently low norm. Jiang, Zhou and Zhu (2025), whom the anchor study cites as finding positive alignment between layer contributions in vision transformers, in fact measure the similarity of hidden representations across layers and find it positive and increasing as layers get closer. Their result is about states.

So the observation that adjacent states are nearly parallel and the observation that adjacent writes anti-correlate are both what one would expect if each layer adds a small vector to a large sum and the next layer takes some of it back. Neither result can be cited against the other.

4.2 Three claims under the name "iterative inference"

The phrase "iterative inference" is used for three claims that the cancellation result treats differently.

The first is convergence: the state moves through depth toward a final representation, so that a readout at layer L gives a progressively better approximation of the final output. Brothers et al. (2024) report that the trajectory of a token's residual embedding follows decreasing loss and that the rate of convergence tracks the model's uncertainty; Belrose et al. (2023) train a per-layer affine translator that decodes every hidden state into a distribution over the vocabulary and find it more predictive and reliable than the logit lens; Yom Din et al. (2023) find that linear maps from intermediate to final representations approximate the final layer far better than reading hidden states directly in the final layer's space, and that with them GPT-2 and BERT often already predict their final output in early layers. Convergence is a claim about states and about the output, and layer

correction does not contradict it.

The second is refinement in the original sense of Greff, Srivastava and Schmidhuber (2016): successive layers revise estimates of the same features rather than computing new ones. Jastrzebski et al. (2017) formalise this for residual networks by showing that residual blocks tend to move features along the negative gradient of the loss. A step along a gradient can subtract as easily as it adds. Refinement in this sense predicts some corrections, and Rushing and Nanda (2024) make the connection explicit: they present an "Iterative Inference hypothesis" as a framework that would make self-repair unsurprising, and cite as evidence heads in GPT-2 Medium whose direct effect falls in proportion to how much of their own output is added back into their input - self-repressing heads, whose graded response to an amplified earlier write resembles the one the anchor study reports for whole layers.

The third is enrichment: layers add information about a token that later layers read, and the added information stays available. Geva et al. (2023) describe early MLPs enriching the last subject token with many subject attributes that attention later extracts; Geva et al. (2022) describe each MLP update as an additive update to a distribution over the vocabulary, decomposable into sub-updates that promote concepts. This is the reading the anchor study contests, and it is the reading that single-layer feature extraction relies on most. But even enrichment is not a claim that every write persists. It is a claim that the writes that matter persist, and the anchor study's own Jacobian evidence - a corrected subspace and a reinforced subspace among the directions that carry most of a write, with the rest of the write outside the directions analysed - is compatible with that.

The tension, then, is not between iterative inference and layer correction. It is between layer correction and a working practice that treats every component of a write, or every latent active in a state, as something the model has added and will keep.

4.3 Where erasure sits in depth

Csordas et al. (2025) also measure a quantity between the two discussed above: the cosine between a sublayer's output and the residual it is added to. Under their reading, negative values mean the sublayer erases part of what is already in the stream and positive values mean it strengthens existing features. In Llama 3.1 70B, after a first layer near zero and a short positive phase, the rest of the first half of the network mostly erases; around the middle a sharp transition occurs, after which layers mostly strengthen what is there, their contributions shrink relative to the stream, and skipping them has little effect on later computation. Patrawala et al. report that the correction is strongest in the first two-thirds of each model's layers. The two studies use different models, different quantities and different thresholds, so their depth profiles cannot be overlaid, but they agree on the shape: subtraction is concentrated in the early and middle part of the network, where interpretability work

usually looks for features, and less present in the late part, where the output distribution is being sharpened.

This matters because it makes the problem worst where the readouts are most used. Lad et al. (2024) find that probe accuracy for context-dependent word meaning peaks in intermediate layers across the families they test, and note that earlier probing studies show the same rise, peak and decline; sparse dictionaries are routinely trained at those depths.

5. Four kinds of non-persistence

Content written into the residual stream can stop being decodable for at least four reasons that the literature documents separately. They license different conclusions about a readout taken before the content disappeared.

5.1 Rejection

The propose-and-reject hypothesis describes content that was wrong for the context: a layer writes several candidates and the next layer removes those that do not fit. The anchor study's example is the token "202" inside "1,808,202", for which a layer might propose magnitudes such as hundreds, thousands and millions, of which context supports only one. If rejection is what happens, a readout taken between proposal and rejection is evidence of presence and nothing more. It is not evidence of use, since by hypothesis the rejected candidate does not reach later computation, and it is evidence against endorsement. The rejection reading is the least established of the four: it is the anchor study's interpretation, supported by the graded and subspace-selective character of the correction and by its concentration on context-dependent tokens, but not by a test of what the rejected directions encode.

5.2 Consumption

Content can be removed because it has been used. Feucht et al. (2024) train linear probes to recover the current and preceding tokens from hidden states of Llama-2-7b and Llama-3-8b and find a sharp "erasure" at the last token of multi-token words and named entities: information about the constituent tokens is lost in early layers, while for other tokens it fades gradually. They read the effect as the footprint of a process that assembles a lexical item from its pieces and then discards the pieces, and use it to read out an implicit vocabulary. The anchor study raises the same possibility for its own effect - that layers may erase information because it is only transiently useful, or to free capacity - but notes that layer correction is more constrained than general erasure, since the reversal comes in the very next layer, leaving only that layer to use what was written. Under consumption, a readout before removal is evidence of presence and may be evidence of use; it is not evidence that the content persists, and its disappearance is not evidence that it was wrong.

5.3 Regulation

Some components oppose a signal as their ordinary job, whatever its content. McGrath et al. (2023) find late MLP layers in a 7-billion-parameter Chinchilla model that lower the logit of the most likely token; at layer 23 of 32, 92 percent of the variance in the downstream change in direct effect after an attention ablation is explained by the compensating response. McDougall et al. (2023) describe a GPT-2 Small head, L10H7, that suppresses tokens which earlier components predict and which appear earlier in the context, explain 76.9 percent of its impact, and find that it accounts for 39 percent of self-repair on a narrow task. Stolfo et al. (2024) find entropy neurons that write into a null space of the unembedding to change the residual norm, and so the confidence of the output, with little direct effect on any logit; Gurnee et al. (2024) find among universal GPT-2 neurons families that change the entropy of the next-token distribution or predict that the next token will not be in a set. Lad et al. (2024) count prediction and suppression neurons across 16 models and find suppression neurons dominating the final layers; in the most capable Phi models they report nearly 15 percent prediction and 25 percent suppression neurons per layer. Ahmad, Seth and Sankarapu (2026) show that the apparent repair after an ablation follows an affine law in the strength of the intervention whose slope is fixed and present without any ablation; of 81 downstream directions examined in Gemma, Qwen, Llama and Mistral models, 68 pass their test, 52 of them counterweights with a negative slope and 16 relays with a positive one, and in the GPT-2 Small indirect-object circuit all seven heads that follow the law are counterweights. Figure 1(a) shows these counts.

Regulation is relevant to layer correction in two ways. It shows that pushing against an upstream signal is a common and well-documented function in transformers, including in GPT-2, a family in which the anchor study finds no layer-level correction. That is a caution about reading the absence of the layer-aggregate signature as absence of correction at the grain of heads and neurons. And it gives a reading of the anchor study's adaptivity result that does not require content-specific rejection: a unit with a fixed negative gain on some direction will oppose more of that direction when more is written and less when less is, which is the graded response Patrawala et al. report. The two readings make different predictions about content. Under rejection the opposed directions should depend on context and encode wrong candidates; under regulation they should be fixed and encode signal strength, for example the confidence of a prediction. The anchor study's evidence that correction is more frequent on context-dependent tokens and later in the context favours a content-sensitive component, but does not separate the two.

For a readout, regulation implies that a feature's value at layer L may be deliberately damped downstream without being wrong or consumed. Presence and use survive; the magnitude at layer L overstates the magnitude the model

acts on.

5.4 Re-encoding

Finally, content can persist while its representation changes, so that a readout fixed to one basis loses it. Belrose et al. (2023) report that the logit lens - decoding any layer with the model's final unembedding - fails to give plausible predictions for BLOOM and OPT 125M and, in their GPT-Neo-2.7B example, before layer 21, and that even where it works its output is hard to read because of "representational drift": features may be represented differently at different layers. They document that the covariance of hidden states drifts apart as the number of layers between them grows, and train per-layer translators to correct for it. Wendler et al. (2024) follow the final-token state of Llama-2 models through depth on non-English prompts and find three phases: far from any output token, then decodable as the right concept but closer to its English form, then moved into the region of the input language. A lens reading of the middle layers says "English"; the final output is not in English. Halawi, Denain and Steinhardt (2023) find that, given incorrect few-shot demonstrations, decoded predictions match those for correct demonstrations up to a "critical layer" and then degrade, and identify late heads that copy false information from the context as a possible cause, since ablating them reduces the effect. The correct answer was present and decodable, and was then overwritten.

Under re-encoding, disappearance at a fixed readout is not disappearance from the model, and a single-layer readout is weaker evidence of endorsement than it appears. The Wendler and Halawi cases also show that a middle-layer readout can be both present and decodable and still not what the model eventually does.

5.5 What the four have in common

All four imply the same thing for practice: presence at depth L does not license persistence, use or endorsement without a further measurement. They differ in what the further measurement should be. Rejection calls for checking whether the feature survives the next layer; consumption for checking whether something downstream read it before it vanished; regulation for checking the magnitude the model acts on; re-encoding for reading later layers in their own basis. Layer correction is the strongest of the four in one respect - it operates on nearly every token, at nearly every early and middle layer, in most current open model families - and the weakest in another: of the four, it is the one whose content has not been examined.

6. What the persistence evidence measures

A second body of work traces features across layers directly. Read together, it supports a narrower conclusion than either side of the debate tends to draw.

6.1 Matching dictionary directions across layers

Balagansky, Maksimov and Gavrilov (2024) align the latents of separately trained sparse autoencoders at different layers of Gemma 2 2B without data, by matching their folded encoder and decoder parameters, and ask a language model to judge whether matched latents mean the same thing. Starting from layers 10, 12 and 20, judged similarity declines gradually but remains significant for about five layers. Laptev et al. (2025) extend the approach with a data-free cosine comparison between a residual latent at one layer and latents in the previous residual stream, MLP and attention outputs, classify each latent by where it most likely came from, and find in Gemma 2 2B that after about layer 18 fewer new latents appear and most are carried forward from earlier layers. Deactivating a latent by switching off its matched predecessor succeeds about 65 percent of the time with a single top cosine match and 73 percent with an exhaustive search. Paulo et al. (2024) find that the natural-language explanations of autoencoders trained on nearby residual layers are highly similar, and Ghilardi et al. (2024) train one autoencoder for each group of contiguous, similar layers in Pythia models with little loss in reconstruction or interpretability.

These results show that directions are reused across depth: a direction a dictionary finds at layer L is likely to have a counterpart at $L+1$ and $L+2$. They do not show that a given token's feature at L is still present at $L+1$. Decoder similarity is a property of the dictionary, aggregated over every context in which a latent fires, and a proposal that is withdrawn on one token can share its direction with a feature that persists on another.

6.2 Following a token's latents through depth

Two studies ask how many layers a latent's activity actually depends on, and both find less persistence than the residual sum would naively suggest. Lawson et al. (2024) train a single sparse autoencoder on residual activations from every layer of Pythia models (and of Gemma 2 2B, Llama 3.2 3B and GPT-2 Small). They expected latents to switch on at some depth and stay on; instead, for a given token or prompt, individual latents are often active at a single layer, while the layer at which a latent fires varies between tokens. The variance of a latent's layer distribution is about two orders of magnitude larger when pooled over tokens than within a token. Applying tuned-lens translators to correct for representational drift leaves the picture qualitatively the same, and larger models show more multi-layer activity, consistent with their adjacent states being more similar. Demou et al. (2026) report the same pattern for crosscoders, dictionaries trained jointly across layers: although the decoder weight of a latent is spread evenly over layers, masking one layer's input at a time shows that each latent's activation depends, on average, on only one or two layers (a mean functional coherence of about 2 in Pythia-410M).

These are the results that come closest, outside the anchor study, to showing per-token transience, and

Figure 1(b) places them next to it. But Demou et al. treat the layer-locality as a pathology of the dictionary objective rather than a property of the model: unconstrained cross-layer parameters let a latent take its value from whichever layer is most convenient. With low-rank factorised weights and a penalty for latents that collapse when one layer is masked, their mean coherence roughly doubles to about 4, probing performance improves by 10 to 30 points of F1, and the latents that span layers are the more interpretable ones. Whether Lawson et al.'s single-layer latents reflect transient content, a training pressure of the same kind, or residual changes of basis that a tuned lens does not undo cannot be told from their results.

6.3 Measuring transport

Three recent methods test persistence more directly. Szablewski and Masiak (2025) fit linear maps from a residual state to the state k layers later, evaluate them in the space of a downstream autoencoder, and use them to distinguish latents that are linearly carried forward from latents synthesised by nonlinear computation; they report an upper bound on transport efficiency and the size of the subspace involved. Poduval et al. (2026) argue that residual activations are so strongly coupled across depth that separately trained per-layer dictionaries spend capacity on the same carried-forward information, and train each later-layer dictionary only on the part of the activation that an affine map from an earlier layer does not explain. Garg et al. (2025) build a cross-layer bottleneck on the premise that the residual stream mixes and duplicates features across adjacent layers. All three treat carried-forward content as the bulk of the stream.

Droste et al. (2026) add a caution about how links are found. In Pythia-160M they build an atlas of cases where a residual feature at layer 7 and an MLP-output feature combine to produce a residual feature at layer 8, validate each by ablating the update feature, and find 38,125 strong transitions; in 88.0 percent of them both decoder cosines (state to target and update to target) are below 0.7, so the decoder-similarity matching used in Section 6.1 would have missed them. A partial check in Gemma-3-4B gives 53.6 percent, which the authors call "directionally consistent with Pythia, but weaker", because the update-to-target cosine alone recovers many of the strongest Gemma effects: a top-30,000 baseline on that cosine finds 68.2 percent of the strong-effect candidates and misses 31.8 percent. If many real transitions are invisible to decoder matching, matching studies undercount both what changes and what persists.

6.4 Reading the two bodies together

The persistence literature does not answer the question the anchor study raises, because it mostly asks a different one. Decoder matching answers whether directions are reused across depth, and the answer is yes, over a horizon of a few layers. Per-token dictionaries answer whether a token's latents stay active, and the answer is often no, but the dictionaries that give that

answer are known to produce layer-locality as an artefact. Transport methods assume that most of the state is carried forward, which the residual arithmetic of Section 4 guarantees for the state, and are silent about the share of each layer's write that survives the next layer. That share is the quantity layer correction concerns. Nobody has measured it at the grain of features.

7. Deletion and swapping as a coarse test of propose-and-reject

A third body of evidence has not been brought to bear on layer correction, although it bears on it directly. Lad et al. (2024) delete single layers, or swap adjacent ones, and report that models keep 72 to 95 percent of their top-1 prediction accuracy without any fine-tuning; their main layer-by-layer intervention results are for four GPT-2 and four Pythia models, and their stage analysis extends to Phi, Llama 3.2 and Qwen 2.5 models. The first and last layers are fragile; the middle layers are robust to both interventions, and swapping two adjacent middle layers does less harm than deleting one. Sun et al. (2024b) find that the middle layers of pretrained transformers share a representation space to a surprising degree and that some tasks tolerate skipping layers, running them in a different order, or running them in parallel. Their main model is Llama2-7B; in an appendix on Mistral-7B and Pythia-6.9B they report that "Mistral follows a remarkably similar trend as Llama2-7B", while "Pythia is less robust to modifications, especially modification of the Layer order". Gromov et al. (2024) remove up to about half of the layers of open models from the Llama-2 and Qwen families, Mistral-7B and Phi-2, chosen by the similarity of representations across layers, with little loss on question answering, and on MMLU even without the small amount of healing fine-tuning they apply; Men et al. (2024) prune layers ranked by a similarity-based influence score. Veit, Wilber and Belongie (2016) showed earlier, for residual image networks, that paths through the network behave like an ensemble whose members do not depend strongly on one another.

Take the propose-and-reject hypothesis literally. If layer i proposes candidates and layer $i+1$ rejects the wrong ones, then swapping them makes layer $i+1$ run first, with nothing to reject, and leaves layer i 's proposals in the stream with no adjacent rejecter. Deleting layer $i+1$ does the same. The hypothesis therefore predicts that, in a model with strong adjacent-layer correction, swapping or deleting the second member of a correcting pair should do more damage than the same intervention on a pair without correction, and more damage than in a model family without the effect.

Lad et al.'s results cannot show this, because they were not designed to and because their main layer-by-layer results come from GPT-2, in which the anchor study finds no correction, and Pythia, which it did not examine. A coarse, family-level version of the comparison is nevertheless already published. Sun et al.

(2024b) find middle-layer skipping and reordering cheap in Mistral-7B, a member of one of the five families in which Patrawala et al. report the correction, and less cheap in Pythia-6.9B. Gromov et al.'s pruning of Mistral-7B and Qwen-family models points the same way; on MMLU the robustness holds with or without their healing fine-tuning, which gives only modest improvements there (healing matters more for BoolQ and for perplexity). Read naively, this is the opposite of what propose-and-reject predicts, and three readings fit it now. Uncorrected proposals may be absorbed by later layers, as the self-repair literature would predict (McGrath et al., 2023; Rushing and Nanda, 2024). They may matter little for the top-1 prediction, the metric the robustness results use, while mattering for features that a dictionary would label. Or the adjacent correction may be less specific to the adjacent pair than "layer $i+1$ corrects layer i " implies; the sublayer analysis already shows MLPs opposing both the preceding MLP and the preceding attention output. No study has selected adjacent layer pairs by their measured correction or scored internal features after the intervention. The comparison that would separate these - swap damage for correcting and non-correcting pairs, in models with and without the effect, scored on internal features as well as on the output - has not been run. Section 12 states it as Study 2.

8. The three proposed consequences, checked against the record

Patrawala et al. connect layer correction to three open problems in feature-based interpretability. Each connection is a reasonable conjecture. For each, the record contains a competing explanation with at least as much support, and in one case a result that sits awkwardly with the proposal.

8.1 Low specificity of dictionary features

The proposal is that an autoencoder trained on one layer cannot separate persistent features from proposals about to be withdrawn, so it will learn latents that fire on content the model is about to correct - misfires that a human or language-model judge would see as irrelevant. This predicts that misfires should be more common at layers and on tokens where correction is active.

The competing explanations are better documented. Chanin et al. (2024) show that when features form a hierarchy, the sparsity objective leads a seemingly monosemantic latent to fail to fire where it should, its role absorbed by more specific child latents, and that changing the dictionary's size or sparsity does not fix this; absorption degrades both recall and the fit of a description. Leask et al. (2025) show that dictionaries are neither complete nor atomic: larger dictionaries contain latents that a smaller one lacks, and many large-dictionary latents decompose into combinations of smaller-dictionary latents. Sun et al. (2025b) show that dense latents, which fire on a large share of tokens and would score as unspecific, are often functional, forming

antipodal pairs that reconstruct specific residual directions tied to position, context binding or entropy regulation. Paulo et al. (2024) document pitfalls in existing ways of scoring explanations and find that an intervention-based score credits latents that the existing scores miss, so that measured specificity depends on the scoring method. Any of these would produce low measured specificity without layer correction.

The proposal and its competitors make different predictions. Absorption predicts that specificity failures concentrate in hierarchical concept families and do not depend on depth; layer correction predicts a dependence on depth (stronger in the first two-thirds of layers), on token type (stronger on numbers, punctuation and newlines) and on model family (absent in GPT-2 and Phi). None of these dependences has been measured with the anchor study's correction counts in hand.

8.2 Large steering coefficients

The proposal is that steering by adding a feature direction must overcome the next layer's correction, which the anchor study's appendix finds weakens only when the earlier write is scaled by a large factor, and that this is why steering by dictionary features has needed amplification up to ten times a feature's maximum observed activation.

The record contains three alternatives and one awkward result. The awkward result is that large coefficients are needed in a family in which the anchor study reports no layer correction. Turner et al. (2023), steering GPT-2-XL by adding the difference between the activations for "Anger" and "Calm", find the vector effective at an injection coefficient of +10 at layer 20, which they note is nearly ten times the norm of the residual stream at the injection site; by their heuristic reading, about 90 percent of the residual after layer normalisation is then set by the steering vector rather than by the prompt. A coefficient of +1 changes the output distribution less than a random vector of the same norm. The anchor study's own Figure 1 shows GPT-2 XL among the models without the adjacent-layer bands, so ten-fold amplification arises in a model in which the layer-level pattern was examined and not found. The two "ten times" figures are on different scales - Turner et al.'s is a multiple of the residual stream's norm at the injection site, the one the anchor study borrows is a multiple of a latent's largest observed activation - so only the qualitative point carries over.

The first alternative is direction quality. Wu et al. (2025) compare steering and concept-detection methods on Gemma-2-2B and 9B in a common benchmark and find that prompting outperforms every representation-based steering method, that difference-in-means directions are the best detectors, and that sparse autoencoder latents are not competitive on either task; Chalnev, Siu and Conmy (2024) note that the effects of steering by an autoencoder latent are hard to anticipate and build vectors that target the intended latents while limiting side effects. If dictionary latents are poor steering

directions, they will need large coefficients whatever the next layer does. The second alternative is that a direction estimated from the model's own activations steers at its natural scale. Arditi et al. (2024) add the unscaled difference between mean activations on harmful and harmless instructions at a single layer and induce refusal of harmless requests across the chat models they study, without amplification. The third is scale. Residual norms grow rapidly with depth (Csordas et al., 2025), and the output variance of pre-normalised transformers grows exponentially with depth (Sun et al., 2025a), so a vector of fixed norm is a shrinking share of the stream deeper in the network, and a coefficient that looks large in absolute terms may be modest relative to the stream it is added to. This third alternative does not explain the GPT-2-XL case, where the effective coefficient is large relative to the stream itself.

Layer correction may still contribute. The prediction that would isolate it is that, at a fixed relative norm and for a fixed direction, the coefficient needed to move behaviour is higher at layers and in families where correction is active, and lower for directions the next layer's Jacobian reinforces. The anchor study proposes the second half of this as future work.

8.3 Transcoders against sparse autoencoders

The proposal is that transcoders, which predict a sublayer's output from its input, can represent a correction as a function of what came in and so separate it from new content, which an autoencoder on the output cannot. Paulo, Shabalin and Belrose (2025) compare transcoders and autoencoders trained on the MLP outputs of the same models on the same data and find transcoder latents more interpretable, and report that their skip-transcoder variant scores higher on automated interpretability and costs less loss when patched in than autoencoders in every model they study: Pythia 160M and 410M, Llama 3.2 1B and Gemma 2 2B. Dunefsky, Chlenski and Nanda (2024) report transcoders at least on par with autoencoders in sparsity and faithfulness on GPT-2 Small, Pythia-410M and Pythia-1.4B, and in human interpretability on a Pythia-410M layer.

The anchor study did not examine Pythia, and the one GPT-2 comparison in the record concerns sparsity and faithfulness rather than interpretability, so whether the transcoder advantage appears in a model without layer correction cannot be read from the record. The comparison that would test the proposal is direct: the size of the transcoder advantage should track the strength of correction at the layer, be largest for MLPs (which the anchor study identifies as executing most of the correction), and shrink or vanish in GPT-2 and Phi. Paulo et al. themselves offer a simpler framing that their design does not exclude. They write that autoencoders "focus on compressing intermediate activations rather than modeling the functional behavior of network components", and that "for transcoders the latents should relate more to concepts necessary for processing the input". A latent tied by its objective to a pattern in the

sublayer's input may by itself have more coherent activating contexts, with or without any correction downstream. Their result does not discriminate between the two accounts.

8.4 What the three checks show

None of the three consequences is refuted, and none is established. In each case the competing explanation is supported by direct measurements, the layer-correction account by an analogy, and the observation that would separate them is a dependence on depth, token type or model family that no study has yet measured. The anchor study's three connections are best read as a list of predictions.

9. Consolidated evidence

Table 1 collects the published values the argument rests on, one row per result, with the model, the quantity measured and the side of the argument the value bears on, so that rows of different kinds are not read as one series. Figure 1 plots the subset of results reported as counts of units or as spans of layers.

Figure 1 makes two points that the table states less directly. Panel (a) shows that where studies count the units or directions that push against an incoming signal and those that push with it, the opposing ones are the majority in each count reported; regulation is not a curiosity. The counts do not estimate a model-wide share: Ahmad et al. call their 52-to-16 split "a ratio specific to a bench selected for legible poles", and the Patrawala bar counts eigen-directions of one Jacobian, split by the sign of the eigenvalue, not units. Panel (b) shows that the persistence horizon depends on what is measured and how: about one layer for the corrected part of a write and for per-token dictionary activity, two to four layers for crosscoder latents depending on the training objective, about five for matched directions. The figure is not a meta-analysis. The bars come from different models and different measures, and they are plotted together to show that the published answers to "how long does a feature last?" differ by measurement, not to estimate a single horizon.

10. A persistence check before a single-layer claim

Algorithm 1 states as a procedure what Sections 3 to 8 argue: which of the five claims a readout at one depth licenses, and what has to be measured before it licenses more.

Three features of the procedure deserve comment. Line 2 is cheap: whether a dictionary was trained on states or on writes changes what its latents can be evidence for, and the anchor study's result is about writes. Lines 4 and 5 require reading later layers in their own basis because the re-encoding evidence (Belrose et al., 2023; Wendler et al., 2024) shows that a fixed readout loses content that persists. Lines 6 to 10 classify a disappearance rather than treating it as one thing, because the four kinds of non-persistence in Section 5 license different

conclusions; in particular, a feature that vanishes after one layer is not thereby shown to have been a rejected proposal. Line 13 makes the model family part of the claim. If a readout's interpretation depends on whether written content survives, the reader should know whether the model is one in which adjacent writes have been measured to oppose each other.

The procedure is demanding, and the author expects that many single-layer findings, as published, would stop at line 1 or line 11. That is the point: the license a readout carries is set by the measurements made, and the author argues that a common pattern is to measure presence while narrating endorsement.

11. The strongest case against this reading

The strongest objection is that this paper makes layer correction look less consequential than it is by reading it as a statement about writes while practice is about states. A residual-stream autoencoder trained on the state after layer L sees layer L 's proposals in full; those proposals are withdrawn only in the state after layer $L + 1$. On this view the state at every depth contains the current layer's not-yet-corrected proposals, exactly as the anchor study says, and every state-based dictionary is exposed. The arithmetic of Section 4 shows that the state as a whole persists; it does not show that the features a dictionary extracts from the state persist, because a dictionary latent can be carried almost entirely by the small, recently written part of the state.

The objection is correct, and it sharpens rather than defeats the argument. It shows that the relevant quantity is the share of a state's dictionary activity carried by the most recent write and withdrawn by the next, a number nobody has reported. Lawson et al.'s per-token result, mainly from Pythia, a family the anchor study did not examine, with checks on Gemma 2 2B, Llama 3.2 3B and GPT-2 small, is the closest evidence that the share is large; Demou et al.'s masking result is the closest evidence that some of that apparent share is produced by the dictionary objective. The correct summary is not that layer correction is harmless for state-based dictionaries, but that its harm is unmeasured, and that the measurement is straightforward with the anchor study's Jacobian method and a trained dictionary.

A second objection is that the competing explanations in Section 8 are not rivals at all, since several mechanisms can contribute to one phenomenon. This is true, and the paper does not claim that layer correction contributes nothing to low specificity, large steering coefficients or the transcoder advantage. It claims that the record does not yet show that it contributes anything, because the observations that would show it - dependence on depth, token type and family - have not been made, while the competitors have direct support.

A third objection concerns the GPT-2 steering result. The anchor study's Figure 1 shows no adjacent-layer bands in GPT-2 XL, but heads that oppose earlier writes are documented in smaller GPT-2 models (copy suppression

Table 1. Published results bearing on whether a layer's write persists, by source. "State" is the residual sum after a layer; "write" is what one layer adds. Values are in each source's own metric and are reported, not recomputed.

#	Source	Model(s)	Quantity	Reported value	Bears on
1	Patrawala et al. (2026)	5 of 7 families	Cosine of adjacent writes	about -0.2; -0.3 to -0.15; min -0.4	Writes oppose
2	Patrawala et al. (2026)	Llama 3.1 8B	Jacobian eigenvalues, correcting pairs	modes near -1, +0.5; ~3000 of 4096 < 0	Selective
3	Patrawala et al. (2026)	OLMo 7B	Pattern over training	none at 5%; present by 21%	Learned
4	Patrawala et al. (2026)	GPT-2, Phi	Adjacent-write pattern	absent	Scope
5	Csordas et al. (2025)	Llama 3.1 70B	Cosine of write with state	negative in much of first half	Erasure early
6	Csordas et al. (2025)	Llama 3.1 70B	Cosine of neighbouring states	often close to 1	States persist
7	Razzhigaev et al. (2024)	GPT, LLaMA, OPT, BLOOM	Procrustes, sequential layers	0.99	States persist
8	McGrath et al. (2023)	Chinchilla 7B	Downstream variance explained, layer 23	92%	Compensation
9	Rushing and Nanda (2024)	GPT-2, Pythia, Llama	LayerNorm share of self-repair	about 30%	Scale
10	McDougall et al. (2023)	GPT-2 Small	Self-repair from copy suppression	39%	Regulation
11	Ahmad et al. (2026)	Gemma, Qwen, Llama, Mistral	Counterweights / relays	52 / 16 of 68	Regulation
12	Ahmad et al. (2026)	GPT-2 Small (IOI)	Counterweight heads	7 of 7	Regulation
13	Feucht et al. (2024)	Llama-2-7b, Llama-3-8b	Prior-token probes, multi-token words	erased in early layers	Consumption
14	Lad et al. (2024)	GPT-2, Pythia (main)	Top-1 kept, delete or swap	72-95%	Robustness
15	Lad et al. (2024)	Phi	Prediction / suppression neurons	~15% / ~25% per layer	Regulation
16	Belrose et al. (2023)	GPT-Neo, BLOOM, OPT	Logit lens	fails; drift	Re-encoding
17	Balagansky et al. (2024)	Gemma 2 2B	Matched-latent similarity	~5 layers	Reuse
18	Laptev et al. (2025)	Gemma 2 2B	Latent origin past layer 18	mostly carried	Reuse
19	Laptev et al. (2025)	Gemma 2 2B	Deactivation via predecessor	65-73%	Link
20	Lawson et al. (2024)	Pythia and others	Layer variance, pooled vs per token	~2 orders larger	Per-token transience
21	Demou et al. (2026)	Pythia-410M	Layers driving a crosscoder latent	~2; ~4 masked	Artefact
22	Droste et al. (2026)	Pythia-160M; Gemma-3-4B	Strong transitions with both decoder cosines < 0.7	88.0%; 53.6%	Method
23	Turner et al. (2023)	GPT-2-XL	Coefficient for Anger-Calm	+10 (~10x residual norm)	Steering
24	Arditi et al. (2024)	Chat models	Unscaled mean difference added	induces refusal	Steering
25	Wu et al. (2025)	Gemma-2-2B, 9B	SAE steering and detection	not competitive	Direction
26	Paulo et al. (2025)	Pythia, Llama 3.2, Gemma 2	Skip transcoder vs SAE	transcoder ahead in all	Transcoders

Algorithm 1. Persistence check for a single-layer readout

```

input: model M, readout R fitted at depth L, feature f, intended claim C
1 P <- presence: R decodes f at L above a stated baseline? if not: stop, no claim
2 O <- object read: state (residual sum) or write (one layer's output)?
3 if C = origin and O = state: require write-level readout at L; else downgrade
4 H <- persistence horizon: decode f at L+1..L+k in each layer's own basis
5   (tuned or learned translator; do not reuse R's basis unchanged)
6 if f vanishes by L+1:
7   corrected? project f on next-layer Jacobian; negative eigenvalue -> "rejected"
8   consumed? intervene on f at L; downstream change before L+1 -> "used, then cleared"
9   regulated? f reappears damped, not removed -> "used at lower magnitude"
10  else: label "transient, cause unknown"
11 U <- use: intervene on f at L; measure effect on later components and output
12 E <- endorsement: compare f with final output; report agreement, not assumption
13 family check: does M show adjacent-write opposition? report it with the claim
14 licensed claim <- strongest of {presence, persistence to H, use, endorsement}
15   whose measurement in lines 1-12 passed
output: licensed claim, horizon H, cause of disappearance if any, family flag

```

and counterweight heads in GPT-2 Small, self-repressing heads in GPT-2 Medium), so GPT-2-XL may correct at the level of heads and neurons even though it does not at

the level of layers. If so, the result would be less awkward for the proposal. No study has compared the two levels in one model.

12. Studies that would settle the open part

Five studies would resolve most of what this analysis leaves open. Each uses published methods.

Study 1: proposal lifetime at the grain of features. In a model with strong adjacent-write opposition (Llama 3.1 8B) and one without (a GPT-2 model of matched size where possible), train sparse dictionaries on each layer's write and on the residual state. For every latent active in layer i 's write on a token, measure its projection in the state after layers $i+1$ and $i+2$, and its projection onto the negative-eigenvalue subspace of layer $i+1$'s Jacobian. Report the share of latent activity withdrawn within one layer, by depth, token type and family. This is the missing number in Section 6.4.

Study 2: swapping correcting pairs. Sun et al. (2024b) already show at the family level that middle-layer skipping and reordering are cheap in Mistral-7B, a family with the correction; what is missing is the pair-level comparison. Using the anchor study's matrix to select adjacent layer pairs with strong and with no opposition, swap and delete each, in families with and without the effect, and score both the output (as Lad et al. do) and internal features (as Study 1 does). Propose-and-reject predicts more damage for correcting pairs, concentrated in features rather than in top-1 accuracy.

Study 3: steering thresholds. For directions from three sources - autoencoder latents, difference-in-means vectors and directions chosen in the reinforced subspace of the next layer's Jacobian - measure the smallest coefficient, relative to the residual norm at the injection site, that moves a fixed behaviour, across depths and across families with and without the effect. Layer correction predicts higher thresholds where correction is active and lower ones in the reinforced subspace; direction quality predicts differences between sources at every depth; norm growth predicts that relative thresholds are flat.

Study 4: controls for scale and bias directions. Massive activations - a few coordinates up to 100,000 times larger than the rest, nearly constant across inputs and acting as bias terms - sit at the starting token and, in many models, at the first strong delimiter such as a period or newline (Sun et al., 2024a), and a 2026 study traces them to a single layer from which they propagate nearly unchanged (Shi et al., 2026). Delimiter tokens are among those the anchor study finds most corrected, while the starting token is the least corrected, and Kobayashi et al. (2023) find that in BERT and GPT-2 the contextualisation performed by feed-forward blocks is largely cancelled by the residual connection and the following normalisation layer, the latter acting through a few outlier dimensions. Recomputing the adjacent-write cosine with massive-activation coordinates removed, and separately for delimiter and content tokens, would show whether a share of the opposition is scale regulation in a handful of coordinates rather than content-specific rejection.

Study 5: what the corrected subspace encodes.

Decode the negative-eigenvalue directions of correcting Jacobians with a tuned lens, with transcoder latents and with interventions, on controlled inputs where the wrong candidates are known, such as the anchor study's magnitude example ("1,808,202") or multi-token words where Feucht et al. locate erasure. Rejection predicts that the corrected directions encode context-inappropriate candidates; regulation predicts that they encode signal strength independent of content; consumption predicts that they encode pieces already assembled into a larger unit.

13. What is not known

The share of a layer's write that is withdrawn by the next layer, at the grain of interpretable features, is not known. Whether the corrected directions encode wrong candidates, used-up pieces or signal strength is not known. Whether adjacent-write opposition contributes to the low specificity of dictionary latents, to the size of steering coefficients or to the transcoder advantage is not known; competing explanations for each are better supported. Whether Pythia shows the effect is not reported. Whether models that lack the effect at the layer level lack it at the level of heads and neurons is doubtful, since copy suppression, self-repressing heads and counterweights are documented in GPT-2, but no study has compared the two levels in one model. Whether the persistence horizon of a feature is a property of the model or of the dictionary used to find it is not known; the evidence that crosscoder layer-locality is partly an artefact of the objective shows that the two have not been separated.

14. Limitations

The analysis rests on published results and inherits their scope. The anchor study's numbers come mostly from Llama 3.1 8B and a single corpus, and its three downstream connections rest partly on figures from an industry report, which this paper reports as the anchor study cites them. Most of the persistence results come from Gemma 2 2B, a member of a family the anchor study examined, and from Pythia models, a family it did not examine. The depth profiles of Csordas et al. and Patrawala et al. were measured on different models with different quantities and are compared in shape only. The derivation in Section 4.1 uses an assumption about the relative lengths of adjacent writes that the anchor study does not report, and is labelled as such. The counts in Figure 1(a) are drawn from tasks and methods selected by their authors and do not estimate how common opposition is across all units of a model. For sources cited only for their headline finding, the paper states that finding as the source's abstract gives it.

15. Conclusion

A transformer's residual stream is a sum, and a sum can stay almost still while its terms cancel. The 2026 finding that adjacent layers write partly against each other does

not contradict the finding that adjacent states are nearly parallel, the finding that predictions converge through depth, or the finding that dictionary directions recur across layers. Those results describe the state; the correction describes the writes.

What the correction does undermine is a habit: reading presence at one depth as persistence, use or endorsement. That habit was already exposed by the documented cases of consumption, regulation and re-encoding, and layer correction adds a fourth case that is pervasive in most of the open model families examined and whose content is unknown. The three downstream consequences proposed for it are predictions, not explanations; each has a competitor with stronger direct support, and each could be tested with the anchor study's own instruments. The practical recommendation is modest and immediate. A finding reported at one layer should say whether it was read from a state or a write, should say how long the feature lasts and in whose basis, should say what kind of disappearance it undergoes if it disappears, and should say whether the model is one in which adjacent writes have been measured to oppose each other.

References

- Ahmad, A., Seth, P., & Sankarapu, V. K. (2026). Every Ablation Is a Dose: Counterweights and the Semblance of Self-Repair. arXiv:2610.02173. doi:10.48550/arXiv.2610.02173.
- Arditi, A., Obeso, O., Syed, A., Paleka, D., Panickssery, N., Gurnee, W., & Nanda, N. (2024). Refusal in Language Models Is Mediated by a Single Direction. arXiv:2406.11717. doi:10.48550/arXiv.2406.11717.
- Balagansky, N., Maksimov, I., & Gavrilo, D. (2024). Mechanistic Permutability: Match Features Across Layers. arXiv:2410.07656. doi:10.48550/arXiv.2410.07656.
- Belrose, N., Ostrovsky, I., McKinney, L., Furman, Z., Smith, L., Halawi, D., Biderman, S., & Steinhardt, J. (2023). Eliciting Latent Predictions from Transformers with the Tuned Lens. arXiv:2303.08112. doi:10.48550/arXiv.2303.08112.
- Brothers, G., Mannering, W., Tien, A., & Winder, J. (2024). Uncovering Uncertainty in Transformer Inference. NeurIPS 2024 Workshop on Foundation Model Interventions. arXiv:2412.05768. doi:10.48550/arXiv.2412.05768.
- Chalnev, S., Siu, M., & Conmy, A. (2024). Improving Steering Vectors by Targeting Sparse Autoencoder Features. arXiv:2411.02193. doi:10.48550/arXiv.2411.02193.
- Chanin, D., Wilken-Smith, J., Dulka, T., Bhatnagar, H., Golechha, S., & Bloom, J. (2024). A is for Absorption: Studying Feature Splitting and Absorption in Sparse Autoencoders. NeurIPS 2025. arXiv:2409.14507. doi:10.48550/arXiv.2409.14507.
- Csordas, R., Manning, C. D., & Potts, C. (2025). Do Language Models Use Their Depth Efficiently? NeurIPS 2025. arXiv:2505.13898. doi:10.48550/arXiv.2505.13898.
- Cunningham, H., Ewart, A., Riggs, L., Huben, R., & Sharkey, L. (2023). Sparse Autoencoders Find Highly Interpretable Features in Language Models. arXiv:2309.08600. doi:10.48550/arXiv.2309.08600.
- Demou, A. D., Koromilas, P., Oldfield, J., Panagakis, Y., & Nicolaou, M. A. (2026). fmxcoders: Factorized Masked Crosscoders for Cross-Layer Feature Discovery. arXiv:2605.09438. doi:10.48550/arXiv.2605.09438.
- Droste, H., Adriano, C. M., Korte, K., & Giese, H. (2026). Where Decoder Cosine Similarity Fails for SAE Feature Flow Discovery. arXiv:2609.12591. doi:10.48550/arXiv.2609.12591.
- Dunefsky, J., Chlenski, P., & Nanda, N. (2024). Transcoders Find Interpretable LLM Feature Circuits. NeurIPS 2024. arXiv:2406.11944. doi:10.48550/arXiv.2406.11944.
- Feucht, S., Atkinson, D., Wallace, B., & Bau, D. (2024). Token Erasure as a Footprint of Implicit Vocabulary Items in LLMs. arXiv:2406.20086. doi:10.48550/arXiv.2406.20086.
- Gao, L., Dupre la Tour, T., Tillman, H., Goh, G., Troll, R., Radford, A., Sutskever, I., Leike, J., & Wu, J. (2024). Scaling and evaluating sparse autoencoders. arXiv:2406.04093. doi:10.48550/arXiv.2406.04093.
- Garg, A., Yu, X., Sajjad, H., & Kahou, S. E. (2025). Cross-Layer Discrete Concept Discovery for Interpreting Language Models. Transactions on Machine Learning Research (2026). arXiv:2506.20040. doi:10.48550/arXiv.2506.20040.
- Geva, M., Caciularu, A., Wang, K. R., & Goldberg, Y. (2022). Transformer Feed-Forward Layers Build Predictions by Promoting Concepts in the Vocabulary Space. EMNLP 2022. arXiv:2203.14680. doi:10.48550/arXiv.2203.14680.
- Geva, M., Bastings, J., Filippova, K., & Globerson, A. (2023). Dissecting Recall of Factual Associations in Auto-Regressive Language Models. EMNLP 2023. arXiv:2304.14767. doi:10.48550/arXiv.2304.14767.
- Ghilardi, D., Belotti, F., Molinari, M., Ma, T., & Palmonari, M. (2024). Group-SAE: Efficient Training of Sparse Autoencoders for Large Language Models via Layer Groups. EMNLP 2025. arXiv:2410.21508. doi:10.48550/arXiv.2410.21508.
- Greff, K., Srivastava, R. K., & Schmidhuber, J. (2016). Highway and Residual Networks learn Unrolled Iterative Estimation. ICLR 2017. arXiv:1612.07771. doi:10.48550/arXiv.1612.07771.
- Gromov, A., Tirumala, K., Shapourian, H., Glorioso, P., & Roberts, D. A. (2024). The Unreasonable Ineffectiveness of the Deeper Layers. ICLR 2025. arXiv:2403.17887. doi:10.48550/arXiv.2403.17887.
- Gurnee, W., Horsley, T., Guo, Z. C., Kheirkhah, T. R., Sun, Q., Hathaway, W., Nanda, N., & Bertsimas, D. (2024). Universal Neurons in GPT2 Language Models. arXiv:2401.12181. doi:10.48550/arXiv.2401.12181.
- Halawi, D., Denain, J.-S., & Steinhardt, J. (2023). Overthinking the Truth: Understanding how Language Models Process False Demonstrations. arXiv:2307.09476. doi:10.48550/arXiv.2307.09476.
- Jastrzebski, S., Arpit, D., Ballas, N., Verma, V., Che, T., & Bengio, Y. (2017). Residual Connections Encourage Iterative Inference. ICLR 2018. arXiv:1710.04773. doi:10.48550/arXiv.1710.04773.
- Jiang, J., Zhou, J., & Zhu, Z. (2025). Tracing Representation Progression: Analyzing and Enhancing Layer-Wise Similarity. arXiv:2406.14479. doi:10.48550/arXiv.2406.14479.
- Kobayashi, G., Kuribayashi, T., Yokoi, S., & Inui, K. (2023). Analyzing Feed-Forward Blocks in Transformers through the Lens of Attention Maps. ICLR 2024. arXiv:2302.00456. doi:10.48550/arXiv.2302.00456.
- Lad, V., Lee, J. H., Gurnee, W., & Tegmark, M. (2024). The Remarkable Robustness of LLMs: Stages of Inference? arXiv:2406.19384. doi:10.48550/arXiv.2406.19384.
- Laptev, D., Balagansky, N., Aksenov, Y., & Gavrilo, D. (2025). Analyze Feature Flow to Enhance Interpretation and Steering in Language Models. arXiv:2502.03032. doi:10.48550/arXiv.2502.03032.
- Lawson, T., Farnik, L., Houghton, C., & Aitchison, L. (2024). Residual Stream Analysis with Multi-Layer SAEs. ICLR 2025. arXiv:2409.04185. doi:10.48550/arXiv.2409.04185.
- Leask, P., Bussmann, B., Pearce, M., Bloom, J., Tigges, C., Al Moubayed, N., Sharkey, L., & Nanda, N. (2025). Sparse Autoencoders Do Not Find Canonical Units of Analysis. ICLR 2025. arXiv:2502.04878. doi:10.48550/arXiv.2502.04878.
- McDougall, C., Conmy, A., Rushing, C., McGrath, T., & Nanda, N. (2023). Copy Suppression: Comprehensively Understanding an Attention Head. arXiv:2310.04625. doi:10.48550/arXiv.2310.04625.

- McGrath, T., Rahtz, M., Kramar, J., Mikulik, V., & Legg, S. (2023). The Hydra Effect: Emergent Self-repair in Language Model Computations. *arXiv:2307.15771*. doi:10.48550/arXiv.2307.15771.
- Men, X., Xu, M., Zhang, Q., Wang, B., Lin, H., Lu, Y., Han, X., & Chen, W. (2024). ShortGPT: Layers in Large Language Models are More Redundant Than You Expect. *arXiv:2403.03853*. doi:10.48550/arXiv.2403.03853.
- Patrawala, A., Feng, J., Jones, E., & Steinhardt, J. (2026). LLM Layers Immediately Correct Each Other. *NeurIPS 2025*. *arXiv:2609.07876*. doi:10.48550/arXiv.2609.07876.
- Paulo, G., Mallen, A., Juang, C., & Belrose, N. (2024). Automatically Interpreting Millions of Features in Large Language Models. *arXiv:2410.13928*. doi:10.48550/arXiv.2410.13928.
- Paulo, G., Shabalin, S., & Belrose, N. (2025). Transcoders Beat Sparse Autoencoders for Interpretability. *arXiv:2501.18823*. doi:10.48550/arXiv.2501.18823.
- Poduval, P., Yeung, C., Desai, N., & Imani, M. (2026). ReSAE: Residualized Sparse Autoencoders for Multi-Layer Transformer Interventions. *arXiv:2605.27819*. doi:10.48550/arXiv.2605.27819.
- Razzhigaev, A., Mikhalechuk, M., Goncharova, E., Gerasimenko, N., Oseledets, I., Dimitrov, D., & Kuznetsov, A. (2024). Your Transformer is Secretly Linear. *arXiv:2405.12250*. doi:10.48550/arXiv.2405.12250.
- Rushing, C., & Nanda, N. (2024). Explorations of Self-Repair in Language Models. *Proceedings of the 41st International Conference on Machine Learning, PMLR 235, 42836-42855*. *arXiv:2402.15390*. doi:10.48550/arXiv.2402.15390.
- Shi, Z., Wang, Z., Yang, F., Wang, Q., & Tang, R. (2026). A Single Layer to Explain Them All: Understanding Massive Activations in Large Language Models. *arXiv:2605.08504*. doi:10.48550/arXiv.2605.08504.
- Stolfo, A., Wu, B., Gurnee, W., Belinkov, Y., Song, X., Sachan, M., & Nanda, N. (2024). Confidence Regulation Neurons in Language Models. *NeurIPS 2024*. *arXiv:2406.16254*. doi:10.48550/arXiv.2406.16254.
- Sun, M., Chen, X., Kolter, J. Z., & Liu, Z. (2024a). Massive Activations in Large Language Models. *COLM 2024*. *arXiv:2402.17762*. doi:10.48550/arXiv.2402.17762.
- Sun, Q., Pickett, M., Nain, A. K., & Jones, L. (2024b). Transformer Layers as Painters. *arXiv:2407.09298*. doi:10.48550/arXiv.2407.09298.
- Sun, W., Song, X., Li, P., Yin, L., Zheng, Y., & Liu, S. (2025a). The Curse of Depth in Large Language Models. *NeurIPS 2025*. *arXiv:2502.05795*. doi:10.48550/arXiv.2502.05795.
- Sun, X., Stolfo, A., Engels, J., Wu, B., Rajamanoharan, S., Sachan, M., & Tegmark, M. (2025b). Dense SAE Latents Are Features, Not Bugs. *NeurIPS 2025*. *arXiv:2506.15679*. doi:10.48550/arXiv.2506.15679.
- Szablewski, A., & Masiak, M. (2025). Activation Transport Operators. *arXiv:2508.17540*. doi:10.48550/arXiv.2508.17540.
- Turner, A. M., Thiergart, L., Leech, G., Udell, D., Vazquez, J. J., Mini, U., & MacDiarmid, M. (2023). Steering Language Models With Activation Engineering. *arXiv:2308.10248*. doi:10.48550/arXiv.2308.10248.
- Veit, A., Wilber, M., & Belongie, S. (2016). Residual Networks Behave Like Ensembles of Relatively Shallow Networks. *NIPS 2016*. *arXiv:1605.06431*. doi:10.48550/arXiv.1605.06431.
- Wendler, C., Veselovsky, V., Monea, G., & West, R. (2024). Do Llamas Work in English? On the Latent Language of Multilingual Transformers. *arXiv:2402.10588*. doi:10.48550/arXiv.2402.10588.
- Wu, Z., Arora, A., Geiger, A., Wang, Z., Huang, J., Jurafsky, D., Manning, C. D., & Potts, C. (2025). AxBench: Steering LLMs? Even Simple Baselines Outperform Sparse Autoencoders. *arXiv:2501.17148*. doi:10.48550/arXiv.2501.17148.
- Yom Din, A., Karidi, T., Choshen, L., & Geva, M. (2023). Jump to Conclusions: Short-Cutting Transformers With Linear Transformations. *LREC-COLING 2024*. *arXiv:2303.09435*. doi:10.48550/arXiv.2303.09435.