LLM Layers Immediately Correct Each Other

Arjun Patrawala **Jiahai Feng** **Erik Jones** **Jacob Steinhardt**

University of California, Berkeley
arjunpatrawala@berkeley.edu

Abstract

Recent methods in language model interpretability employ techniques such as sparse autoencoders to decompose residual stream contributions into linear, semantically-meaningful features. Our work demonstrates that an underlying assumption of these methods—that residual stream contributions build additively upon each other—is insufficient to fully explain model behavior. Specifically, we identify the Transformer Layer Correction Mechanism (TLCM), wherein adjacent transformer layers systematically counteract each other's contributions to the residual stream. TLCM appears in 5 out of 7 major open-source model families and activates across nearly all tokens in diverse texts. To understand TLCM, we show that it emerges during pretraining, operates most strongly on punctuation and numbers, and adaptively calibrates its correction strength based on the preceding layer's output. We further show that TLCM actively corrects a small subspace and promotes other subspaces, different from standard model behavior. We advance the "propose-and-reject" hypothesis: layers may propose multiple candidate features, while subsequent layers selectively filter out inappropriate ones. Finally, we discuss how our findings help explain three persistent challenges in feature-based interpretability: why extracted features descriptions often suffer from low specificity; why feature-based interventions for model steering fail at low magnitude; why recent work finds cross-layer transcoders outperform SAEs. [1]

1 Introduction

Mechanistic interpretability aims to understand large language models (LLMs) by dissecting the functions of their components. This research direction has critical implications for monitoring and controlling language models [29, 42, 27, 13, 39], as well as designing architectural improvements [51, 50].

A foundational assumption in many interpretability methods is that transformer layers progressively build upon each other's contributions to enrich representations in the residual stream. This perspective has motivated numerous techniques that extract features from transformer layer outputs, including linear probes [3], the logit lens [37], sparse autoencoders (SAEs) [6], and cross-layer transcoders (CLTs) [4, 12, 40]. Similarly, analyses of factual recall often characterize successive layers as gradually augmenting entity representations with additional recalled information [16, 33, 36, 21].

In this work, we introduce the Transformer Layer Correction Mechanism (TLCM), in which adjacent transformer layers systematically reverse portions of each other's contributions. Specifically, we find that in 5 out of 7 open-weight LLM families (Llama 3, OLMo, Mistral, Gemma, and Qwen2), layer $i+1$ consistently produces contributions that partially oppose those of layer i. TLCM challenges the conventional view that layer contributions primarily add information to the residual stream; instead, layers actively edit the residual stream by selectively promoting and rejecting components

[1] Code is available at `https://github.com/arjunpat/transformer-correction`

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

from previous layers. This suggests that analyzing output features of each layer independently may provide an incomplete or misleading understanding of a layer's function, as these features may be subsequently corrected.

To begin, we characterize TLCM through a series of observational experiments in Section 4. First, we show that TLCM is not present at initialization but emerges gradually during pretraining, suggesting that it is a persistent characteristic of LLM training dynamics. Second, we find that TLCM fires most frequently on tokens with high contextual dependency, including numbers, dates, and punctuation. We hypothesize that TLCM is important for handling tokens with high contextual dependency. Finally, we show that TLCM is the combined effort of both attention and MLP.

In Section 5, we focus our experiments on TLCM's underlying mechanisms. First, we use causal interventions to show that TLCM is adaptive and directly dependent on the prior layer, calibrating its correction strength based on the scale of the preceding layer's output. Then, we demonstrate that TLCM systematically corrects and reinforcing specific components of the previous layer by showing an empirical relationship between the eigenvectors of the transformer layer's Jacobian and the corrected subspace. Finally, we synthesize our findings into the "propose-and-reject hypothesis", which states that models can perform enrichment though a process of proposing multiple potential features and correcting the inappropriate ones.

Our findings are helpful for understanding three persistent challenges in feature-based interpretability: (1) why extracted features descriptions often suffer from low specificity; (2) why model steering interventions require high amplification to be effective; and (3) why cross-layer transcoders outperform sparse autoencoders in recovering interpretable features.

We hope our work establishes a corpus of well-characterized phenomena that provides empirical constraints for theories of language model interpretability, and that our findings inform stronger alignment and model steering techniques.

2 Related Work

Feature-based interpretability. A common interpretability paradigm interprets the role of a layer in a neural network as the features present in its output. The linear representation hypothesis posits that neural networks represent concepts as linear features in their activation space [35, 38]. Feature-based interpretability extracts features using supervised linear probes [3] as well as other more sophisticated techniques that can be unsupervised and/or causal (e.g. nostalgebraist [37], Burns et al. [8], Geiger et al. [15], Bricken et al. [6]).

Correction mechanisms. Some recent works have found self-repair and correction exists in models, in which if components in large language models are ablated, later components will change their behavior to compensate [43]. Self-repair mechanisms have been discovered in varied places: ablations of attention layers with later compensation [25, 32], resilience to swapping transformer layers [25], among others. Other research has found mechanisms that are conjectured to help with self-repair, like copy suppression in which an attention head will decrease the probability of predicting a token that has already appeared in the context [31]. TLCM is more pervasive (occurring multiple times on nearly all tokens) and occurs during normal model operation, not specifically during ablations.

Residual stream characterization. The Iterative Inference Hypothesis proposes that transformers progressively refine their latent representations through successive layers [7]. Other work builds methods that uncovers semantically-meaningful latents found in the residual stream [5]. Another line of work highlights the process of enrichment: transformer layers construct enriched representations in the residual stream to perform next-token prediction. For example, function vectors encode input-output functions and are placed in the residual stream to induce execution [48]. Additionally, models proactively consolidate entity-related information into the residual stream before it becomes relevant for prediction [21, 26, 20, 16, 19, 44].

3 Background

Transformer notation. Large language models (LLMs) convert a token sequence into a probability distribution over subsequent tokens. Input tokens are first embedded in the d_{m}-dimensional residual space with positional embeddings, then passed through n transformer layers before being unembedded

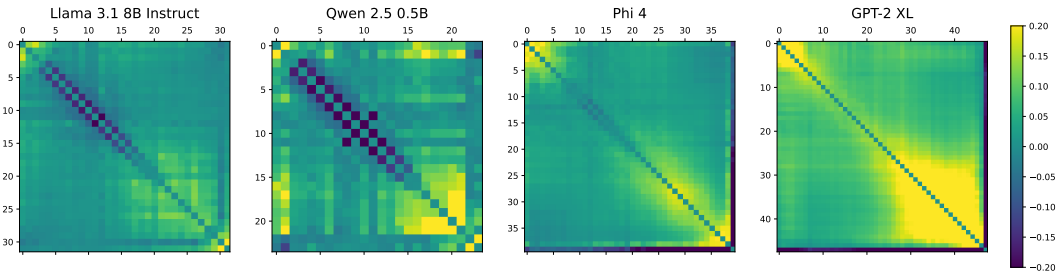


Figure 1: We plot $\text{clamp}(\mathbf{M}, -0.2, 0.2)$ for four models, zeroing diagonal entries. TLCM is visible on the left two plots, characterized by off-diagonal blue stripes of negative cosine similarity, concentrated in the first two-thirds of the model. In contrast, TLCM is not present on the right two plots.

into token logits.

$$\mathbf{x}_0 = \text{Embed}(\text{toks}) \qquad \mathbf{x}_{i+1} = \text{Layer}_\text{i}(\mathbf{x}_i) \qquad \text{logits} = \text{Unembed}(\mathbf{x}_n)$$

Each transformer layer contains attention and MLP sublayers that produce "contributions" to the residual stream:

$$\mathbf{x}'_i = \mathbf{x}_i + \text{Attn}_\text{i}(\mathbf{x}_i) \qquad \mathbf{x}_{i+1} = \mathbf{x}'_i + \text{MLP}_\text{i}(\mathbf{x}'_i)$$

where $\mathbf{x}'_i$ is an intermediate state. The marginal contribution of layer i is defined as $\mathbf{c}_i = \text{Attn}_\text{i}(\mathbf{x}_i) + \text{MLP}_\text{i}(\mathbf{x}_i + \text{Attn}_\text{i}(\mathbf{x}_i))$, making $\mathbf{x}_{i+1} = \mathbf{x}_i + \mathbf{c}_i$. We define $\mathbf{c}_{i,t}$ to be the contribution of the ith layer on token t.

Feature-based interpretability. Recent interpretability methods decompose residual stream contributions using sparse autoencoders (SAEs) to extract meaningful features for a residual stream contribution $\mathbf{c}$ [11, 14, 47]:

$$\mathbf{c} = \mathbf{e} + \sum_{i=1}^{k} \beta_i \mathbf{v}_i$$

where $\mathbf{e}$ is the error, k is a small integer (usually less than 500), and $\mathbf{v}_i$ are the orthogonal feature vectors with activations β_i. These features can be labeled with semantic meanings and manipulated at inference time to steer model behavior. [30, 47, 9].

4 Transformer Layer Correction Mechanism

In this section, we introduce the Transformer Layer Correction Mechanism and conduct a series of observational studies to characterize its functioning. We first define it (Section 4.1), show that it develops during training (Section 4.2), demonstrate how it varies across token types (Section 4.3), and finally demonstrate it occurs via collaboration of both attention and MLP sublayers (Section 4.4).

4.1 Uncovering the Traces

We motivate the discovery of TLCM through a hypothesis about how transformer layers collaborate to enrich the residual stream. Specifically, we hypothesize that layers operate in two distinct modes: (1) contributing novel information to the residual stream, or (2) reinforcing existing information.

To test this hypothesis, we quantify layer interactions using cosine similarity between their contributions. High positive cosine similarity indicates that two layers reinforce similar representations, near-zero similarity suggests their contributions are largely disjoint, and negative cosine similarity implies one layer *reverses* or *corrects* the other's contribution. Consistent with our hypothesis, recent work has found positive cosine similarity between layer contributions in ViTs [24].

Setup. We test this across several open-weight language models. For a given token t, we define the similarity matrix $\mathbf{M}_t$, where $\mathbf{c}_{i,t}$ represents the contribution of the i-th transformer layer (as defined in equation 3):

$$\mathbf{M}_t[i, j] := \text{cossim}(\mathbf{c}_{i,t}, \mathbf{c}_{j,t}) \text{ for } i \neq j$$

We then average these matrices element-wise across approximately 100,000 tokens, across random documents in WikiText [34] using HuggingFace Transformers [49]: $\mathbf{M} = \frac{1}{n}\sum_t \mathbf{M}_t$. These documents include a variety of languages and code.

Results. Plotted in Figure 1, $\mathbf{M}$ reveals a reversing effect, captured by the following observations:

- Across this large corpus, adjacent layers (layer i and $i+1$) on average have opposing contributions, evidenced by their negative cosine similarity which averages ≈ -0.2.
- Curiously, non-adjacent layers (layer i and $i+j$, $j>1$) **do not** on average have opposing contributions; their contributions are predominately orthogonal or positively correlated, consonant with our hypothesis above.

We term this phenomenon—the systematic partial reversal of layer i by layer $i+1$—the Transformer Layer Correction Mechanism (TLCM). We see TLCM across a diverse set of model families including Llama 3 [17], OLMo [18], Mistral [22, 23], Gemma [45, 46], and Qwen2 [52], plotted in Figure 1 and Appendix D. TLCM persists across different text types, model scales, and model categories (instruction-tuned and conversational). The correction mechanism is most pronounced in the first two-thirds of each model's layers. Notably, TLCM is absent in two prominent model families: GPT-2 [41] and the Microsoft Phi models [1, 2], as shown in Figure 1. This absence may stem from their architectures—Phi-3, while based on Llama 2, incorporates dropout blocks after MLP and attention sublayers, similar to GPT-2's design. For the purposes of this paper, we study the TLCM in Llama 3.1 8B ($d_m = 4096$).

We plot the distribution of adjacent layer cosine similarity in Figure 2b, which reveals a strongly bimodal distribution. This suggests that TLCM is a *separate mode of operation*, rather than an idiosyncrasy of normal operation. Based on this plot, we define TLCM as adjacent layer contributions with cosine similarity below -0.1.

Discussion. The presence of highly anti-correlated vectors in such high-dimensional space is striking. To provide a sense of how unlikely that the two outputs of adjacent layers exhibit TLCM (i.e. have cosine similarity less than -0.1), we estimate the probability of this happening under two different null hypotheses in which the outputs are sampled randomly instead.

First, suppose the two outputs are drawn independently from a standard d-dimensional normal distribution. Then, two random vectors in d-dimensional space have expected cosine similarity 0 and variance $1/d$ (Appendix B.1). Therefore, as d scales to 4096 (Llama 8B 3.1), the probability that TLCM occurs for two adjacent layers is as rare as a 6-sigma event.

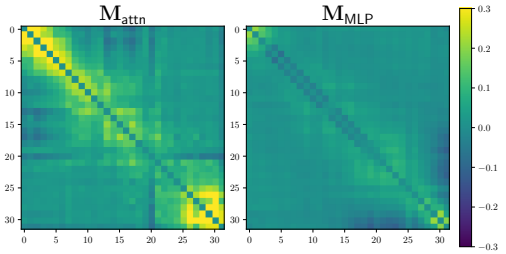
Second, now suppose that the two outputs are drawn independently from empirical distributions of their respective layers. Compared to the earlier normally distributed setting, this accounts for the possibility that the output space of the layers could be low rank or simply be pointing in opposing directions (see Appendix B.2). We this distribution by calculating the empirical mean and standard deviation of cosine similarities between contributions from layers i and $i+1$ across four large documents. Specifically, we sample from the same layer ranges where TLCM is most active ($4 \leq i < 20$) but on adjacent contributions from different tokens. We find that a typical instance of TLCM is approximately as rare as a 3-sigma event. Therefore, TLCM—occurring multiple times on hundreds of thousands of tokens—is exceedingly unlikely to arise from chance alone. There must be a meaningful relationship between the two layers, either through a common confounder or direct dependence. In Section 5.1, we show that this relationship is indeed a direct dependence between adjacent layers.

Intuitively, transformer layers may erase information from the residual stream for various reasons: information may be only transiently useful, or deletion may free capacity for new information. Under either explanation, information written in layer i could be deleted in any layer j where $j > i$. However, TLCM is more constrained: reversal occurs *predominantly* in the directly subsequent layer. This pattern is puzzling from an efficiency standpoint, as it implies written information can only be exploited by a single downstream layer before erasure.

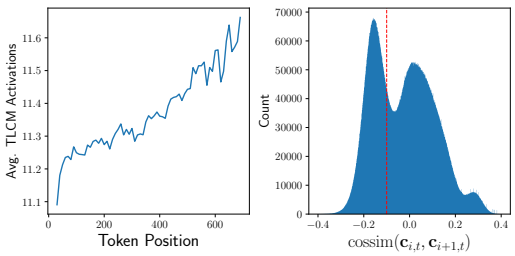
4.2 TLCM Develops During Training

We next investigate how TLCM emerges over training. In particular, TLCM could be:

- An inherent characteristic of transformer architectures, emerging from their fundamental design rather than through learning.
- A transient phenomenon that arises due to, for example, unstable training dynamics.
- A mechanism developed during post-training, either by RL or SFT.



(a) Observe the correlations between attention sublayers, plotted as $\text{clip}(\mathbf{M}_{\text{attn}}, -0.3, 0.3)$ on the left. Traces of TLCM are seen in the plot of $\text{clip}(\mathbf{M}_{\text{MLP}}, -0.3, 0.3)$ on the right.



(b) The left depicts the relationship between average TLCM activations and token position. The right illustrates the distribution of adjacent transformer layer similarities, with the dotted red line at $\text{cossim} = -0.1$.

Figure 2

Setup. To address these possibilities, we study how TLCM develops throughout the training process. We examine seven checkpoints from each of three fully open-source models: OLMo 1B, OLMo 2 1B, and OLMo 2 7B. For each checkpoint, we compute $\mathbf{M} = \frac{1}{n}\sum_t \mathbf{M}_t$ using a standardized block of text.

Results. We find that TLCM emerges over the course of pretraining; in OLMo 7B, for example, it first manifests at training step 135,500 (approximately 21% through the total 651,581 steps), corresponding to roughly 0.5 trillion processed tokens. Appendix Figures A9, A10, and A11 demonstrate that the mechanism's strength increases markedly during the latter two-thirds of training for all three models. This suggests that TLCM is

- *Learning-induced*: TLCM does not appear in untrained models.
- *Persistent*: Despite weight-decay, TLCM strengthens over training and is unlikely to be some training pathology.
- *Pretraining-derived*: TLCM emerges during pretraining, not SFT or RL.

4.3 TLCM Varies by Token

We next study how TLCM activations vary across different token classes.

Setup. We curate 100 realistic queries requesting long-form content about science, technology, philosophy, government, history, and other topics. We then generate responses to these queries using Llama 3.1 8B. For each token t in the response, we count the number of corrections for t as: $|\{i \mid \text{cossim}(\mathbf{c}_{i,t}, \mathbf{c}_{i+1,t}) < -0.1\}|$. This corpus contains 79k tokens. To ensure we identify only systematic patterns, we filter out tokens appearing fewer than 20 times and compute the mean number of TLCM activations for each unique token.

Results. We observe significant variation in correction frequencies across tokens on Llama 3.1 8B; some tokens had few corrections on average, while others had many more. Some qualitative examples of low-correction tokens (< 8 corrections per token on average) include:

> By, workshops, indigenous, vinyl, implications, cognitive, -being, innovative, innovation, regulatory, greenhouse, mindfulness, platforms, trends, therapy, example, learning, iving, planning, inclusive, cloud, classical, proposal, -friendly, sustainability, biodiversity, ting, blog, memo, system, `<|begin_of_text|>`

And some examples of high-correction tokens (> 11 corrections per token on average):

> 202, Jul, 26,]\n, ,\n\n, Today, \n\n, $, Name, State, [, 201, assistant, Address, \t, 4, \n, user, at,]\n\n, Date, , Title,], -, Your, over, Date, %, high,],,)\n, reach, up, share,)**, well, one, D, time,):, take, forward, from, 1, :, access, you, 12, not, [, I, City, need, low, Thank, (, make, 10, look, your, 0, such, *

We list token-level TLCM activation statistics (mean and standard deviations) in Table A1 and A2. We also compute results for Gemma 2 2B Instruct across the same 79K tokens, listing full results in Table A3 and A4.

We find statistically significant differences in the frequency of TLCM activations. For example, the token `\n` averages 12.41 (± 0.1 99% CI) TLCM activations while `sustainability` averages 7.63 (± 0.46 99% CI) on Llama 3.1 8B. Numbers, punctuation, dates, and brackets rank highly for frequent TLCM activations, while standard English terms like `community`, `training`, and `understanding` have less frequent activations. Notably, Llama's `<|begin_of_text|>` token and Gemma's `<bos>` token exhibit the fewest average TLCM activations across the entire corpus.

Discussion. We observe that tokens exhibiting low TLCM correction rates typically possess unambiguous, context-independent semantic meaning. These tokens generally maintain consistent interpretation regardless of their surrounding context.

In contrast, high-correction tokens seem to demonstrate greater average contextual dependency. This category includes: 1) Numbers that form larger parts of numbers, where grasping the complete value of the number requires consolidating information *across* tokens 2) Date-related tokens which require surrounding context for complete temporal reference 3) Punctuation marks (e.g., ':') whose semantic role varies with usage context.

Recent work shows [28] that newline tokens can be used for planning upcoming tokens, a task that requires contextual processing—newline tokens and similar (`\n \n`, `:\n`, etc.) have high rates of TLCM activation. Intuitively, models may aggregate information into concluding punctuation marks because the causal attention mask prevents them from aggregating information in earlier tokens.

Llama's `<|begin_of_text|>` token and Gemma's `<bos>` are particularly revealing, with 0 and 1 TLCM activations respectively. These tokens are *always* first in the context window during training. In this position, correction is unnecessary; the optimal next-token prediction with no context is simply the empirical unigram distribution from the start of training documents, which we suspect obviates complex contextual processing.

To test this conjecture—that TLCM activations relate to contextual processing—we perform a simple check: we plot the average number of TLCM activations per token at different points in the LLM context window; Intuitively, the 1000th token can attend to 999 preceding tokens while the 5th token can only attend to 4, suggesting we should observe higher TLCM activation rates at later positions. Indeed, across 2000 long WikiText documents, we find that every 1000 tokens of additional context corresponds to approximately 1 additional TLCM activation, as demonstrated in Figure 2b.

4.4 Attention and MLPs Alone Do Not Explain TLCM

We next investigate the role of MLP and attention sublayers in TLCM. For instance, TLCM could arise purely from MLP-to-MLP interactions, or MLPs could selectively reverse only attention sublayer contributions. To address these questions, we conduct an analysis similar to Section 4.1, but at the sublayer level.

Setup. We compute a similarity matrix, instead using the attention contribution $\mathbf{M}_{\text{attn},t}[i,j] = \text{cossim}(\text{Attn}_i(\mathbf{x}_{i,t}), \text{Attn}_j(\mathbf{x}_{j,t}))$ and plot the average matrix $\mathbf{M}_{\text{attn}} = \frac{1}{n}\sum_t \mathbf{M}_{\text{attn},t}$. We also plot the average matrix for the MLPs: $\mathbf{M}_{\text{MLP}} = \frac{1}{n}\sum_t \mathbf{M}_{\text{MLP},t}$, with $\mathbf{M}_{\text{MLP},t}[i,j] = \text{cossim}(\text{MLP}_i(\mathbf{x}'_{i,t}), \text{MLP}_j(\mathbf{x}'_{j,t}))$.

Results. We identify three key findings. First, attention sublayers produce positively correlated contributions with other attention sublayers (Figure 2a). Second, MLP sublayers produce anti-correlated contributions with the subsequent layer's MLP, exhibiting a similar pattern to TLCM. Third, MLP contributions are also anti-correlated with the preceding attention sublayer (Appendix D). In summary, MLPs produce contributions that are anti-correlated with both the preceding MLP and attention sublayers, suggesting that MLPs are primarily responsible for *executing* corrections.

Since MLPs partially reverse contributions from both the preceding attention and MLP sublayers, the correction mechanism appears to be a transformer layer-level phenomenon. Additionally, our prior experiments suggest contextual dependency plays a role in TLCM's functioning, something only possible if TLCM operates partially through the attention sublayer. For these reasons, we focus our study of TLCM at the transformer layer level.

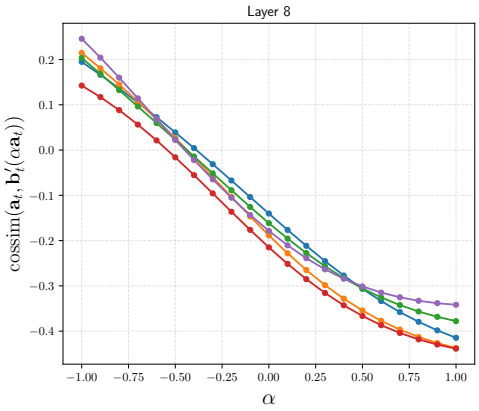


Figure 3: TLCM exhibits a linear relationship between its correction strength and the prior layer's contribution, suggesting that systematic attenuation of the prior layer, shown on 5 randomly sampled tokens. For negative α values, layers demonstrate compensatory behavior, evidenced by positive cosine similarity.

5 TLCM Adaptivity

In this section, we investigate TLCM's underlying mechanistic functioning. In Section 5.1, we show that TLCM is adaptive and directly dependent on the preceding layer's outputs. Then, in Section 5.2, we illustrate a connection between the transformer layer's Jacobian and the subspaces targeted for correction. Using this connection, we show that TLCM selectively targets subspaces for correction, while ignoring or promoting the others.

Finally, we argue for the propose-and-reject hypothesis, which explains the process of enrichment as proposing, checking, and rejecting features in a 2-layer sequence; in this picture, TLCM corrects an "undesirable" component of the previous layer.

Notation. For a particular token t, consider the marginal contribution of layer i and layer $i+1$ to be $\mathbf{a}_t$ and $\mathbf{b}_t$, respectively. Recall that the correction mechanism is currently described by a negative cosine similarity between $\mathbf{a}_t$ and $\mathbf{b}_t$ across a variety of layers and tokens. Additionally, observe that the d_m-dimensional input to layer $i+1$ would be $\mathbf{x}_{i,t} = \mathbf{x}_{i-1,t} + \mathbf{a}_t$. We define a function $\mathbf{b}'_t$ that captures contribution of layer $i+1$ when the previous layer's output is perturbed by $\mathbf{\Delta}$. Specifically:

$$\mathbf{b}'_t(\mathbf{\Delta}) := \text{Layer}_{i+1}(\mathbf{x}_{i,t} + \mathbf{\Delta})$$

Note that $\mathbf{b}'_t(\mathbf{0}) = \mathbf{b}_t$, i.e., the original layer contribution.

5.1 TLCM is Adaptive

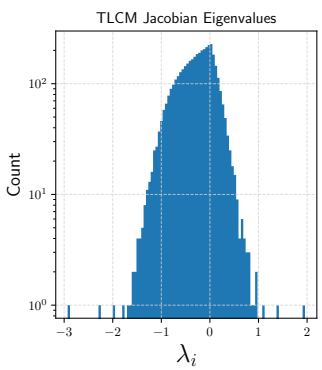
In Section 4.1, we argue that the anticorrelation between the contributions of adjacent transformer layers cannot be explained by randomness—layer i and $i+1$ anti-alignment must be explained by either a common underlying cause or a direct dependence between the layers.

Here, we perform a causal intervention to show that the anti-alignment is caused by a *direct dependence* between the layers. In particular, we find that TLCM's correction is adaptive; as we scale layer i's contribution, layer $i+1$ scales its correction, demonstrating causality.

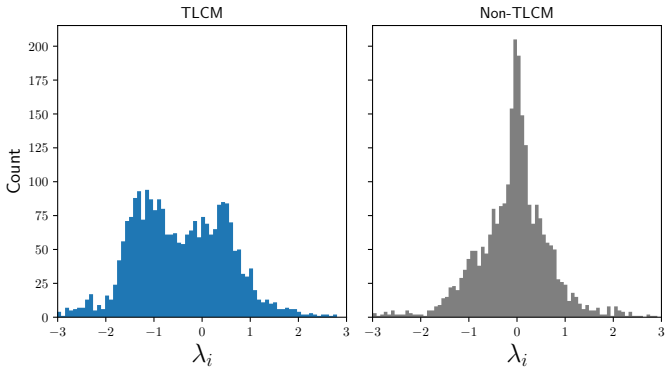
Setup. To understand whether layer $i+1$ is adaptively correcting the contribution of layer i, we add some perturbation $\mathbf{\Delta} = \alpha\mathbf{a}_t$, for $\alpha \in [-1, 1]$ to the input of layer $i+1$. Formally, the input to layer $i+1$ is intervened to become

$$\mathbf{x}_{i-1,t} + \mathbf{a}_t + \alpha\mathbf{a}_t = \mathbf{x}_{i-1,t} + (1+\alpha)\mathbf{a}_t$$

If layer $i+1$ is attuned to correcting $\mathbf{a}_t$, we should expect to see the correction increase as α is scaled up from 0 and decrease as α is decreased from 0. To this end, we measure the cosine similarity between $\mathbf{a}_t$ and $\mathbf{b}'_t(\alpha\mathbf{a}_t)$ at $\alpha \in \{-1, -0.9, \ldots, 0.9, 1\}$.



(a) Nearly 3/4 of the 4096 eigenvalues of the Jacobian $\bar{\mathbf{J}}$ are negative.



(b) Left: Strongest components of $\mathbf{a}_t$ are corrected or reinforced by the TLCM Jacobian. Right: For non-TLCM Jacobians, $\mathbf{a}_t$'s strongest components are predominantly untouched.

Figure 4

Results. As expected, increasing α leads to stronger correction. Decreasing α leads to less correction. Figure 3 depicts this near-linear relationship.

When $\alpha = 1$—effectively doubling $\mathbf{a}_t$ in the residual stream—layer $i + 1$ responds by adjusting its contribution to further oppose $\mathbf{a}_t$. The smooth adaptation we observe indicates layer $i + 1$'s dynamic regulation of the previous layer's contribution. This pattern appears consistently across all layers where TLCM is active, with complete results presented in Appendix E.

Our experiments also reveal a *compensatory* effect. Specifically, in Figure E at $\alpha = -1$, the cosine similarity between the adjacent layers becomes *positive*; layer $i+1$ effectively *boosts* the contribution of layer i. This effect occurs most strongly in earlier layers. This finding aligns with recent research on self-repair mechanisms that enable models to recover from interventions [32]. However, we maintain cautious interpretation: extreme values of α place the model in counterfactual states that do not appear during standard inference or training. Moreover, this compensation effect may stem from self-repair mechanisms unrelated to TLCM.

Discussion. We have demonstrated that, when TLCM is active, there is a causal relationship between layer i and layer $i + 1$'s correction. Adjusting layer i's contribution elicits an immediate response from layer $i + 1$.

This finding is striking for a subtle reason. For layer $i + 1$ to respond sensatively to layer i, it must *identify* which component of the residual stream was added by layer i versus components contributed by earlier layers. This is a challenging task; layer $i + 1$ would need sophisticated mechanisms to isolate layer i's specific contribution. We therefore consider an alternative hypothesis: Layer i's contribution might comprise both a "desirable" and "undesirable" component; layer $i + 1$ simply targets the latter for correction. For instance, layer i might propose n features to the residual stream, after which layer $i + 1$ verifies and removes only incorrect features while leaving the rest intact. This naturally motivates two questions:

1. Does TLCM correct the entire previous layer contribution, or does it target specific components?
2. If TLCM correction is selective, can we identify which subspaces are targeted most aggressively?

5.2 Isolating the Correction Subspace

In this section, we execute an experiment to answer the prior two questions. A key unknown is *how* the previous layer's contribution is attenuated: is it scaled back uniformly, or are specific subspaces selectively targeted for correction?

The layer Jacobian—the local linearization of the both the attention and MLP sublayers—contains information about how the entire layer will respond to input perturbations. By developing tools to characterize and visualize the Jacobian, we can understand which vector directions will be aggressively corrected by the layer versus which directions will remain untouched.

Notation. As previously defined, consider $\mathbf{b}'_t$ to be the function representing transformer layer $i+1$. We can linearize this function by computing the Jacobian $\nabla \mathbf{b}'_t(0) = \mathbf{J} \in \mathbb{R}^{d_{\text{model}} \times d_{\text{model}}}$ of the transformer layer, such that $\mathbf{b}'_t(\mathbf{\Delta}) \approx \mathbf{b}_t + \mathbf{J}\mathbf{\Delta}$. Finally, let $\overline{\mathbf{J}} := \frac{1}{2}(\mathbf{J} + \mathbf{J}^\top)$ be the symmetrized Jacobian.

Claim. The eigenvectors of $\overline{\mathbf{J}}$ with negative eigenvalues correspond to the corrected subspace, with the eigenvalue dictating the correction strength. In a similar vein, we argue that the positive eigenvalue eigenvectors are reinforced proportional to their eigenvalue. Finally, eigenvectors with near zero eigenvalue are mostly untouched by this layer.

Proof. An input perturbation to layer $i+1$ is "corrected" if $\text{cossim}(\mathbf{b}'_t(\mathbf{\Delta}) - \mathbf{b}_t, \mathbf{\Delta}) < 0$. Observe that this condition is true if and only if $\langle \mathbf{b}'_t(\mathbf{\Delta}) - \mathbf{b}_t, \mathbf{\Delta} \rangle < 0$. For a small enough perturbation, the condition is equivalent to $\langle \mathbf{J}\mathbf{\Delta}, \mathbf{\Delta} \rangle < 0$. Because $\langle \mathbf{J}\mathbf{\Delta}, \mathbf{\Delta} \rangle = \langle \mathbf{\Delta}, \mathbf{J}\mathbf{\Delta} \rangle$, the condition becomes:

$$\langle \mathbf{J}\mathbf{\Delta}, \mathbf{\Delta} \rangle = \frac{1}{2}\left(\langle \mathbf{J}\mathbf{\Delta}, \mathbf{\Delta} \rangle + \langle \mathbf{\Delta}, \mathbf{J}\mathbf{\Delta} \rangle\right) = \langle \frac{1}{2}(\mathbf{J} + \mathbf{J}^\top)\mathbf{\Delta}, \mathbf{\Delta} \rangle = \langle \overline{\mathbf{J}}\mathbf{\Delta}, \mathbf{\Delta} \rangle < 0 \tag{1}$$

Moreover, by the spectral theorem, $\overline{\mathbf{J}}$ has an eigendecomposition $\overline{\mathbf{J}} = \mathbf{Q}\mathbf{V}\mathbf{Q}^\top$ with unitary eigenvector matrix $\mathbf{Q} \in \mathbb{R}^{d_{\text{m}} \times d_{\text{m}}}$ and eigenvalues are $\text{diag}(\mathbf{V})$. Plugging this into inequality 1:

$$\langle \overline{\mathbf{J}}\mathbf{\Delta}, \mathbf{\Delta} \rangle = \langle \mathbf{Q}\mathbf{V}\mathbf{Q}^\top\mathbf{\Delta}, \mathbf{\Delta} \rangle = \mathbf{\Delta}^\top\mathbf{Q}\mathbf{V}\mathbf{Q}^\top\mathbf{\Delta} = (\mathbf{Q}^\top\mathbf{\Delta})^\top\mathbf{V}(\mathbf{Q}^\top\mathbf{\Delta}) < 0 \tag{2}$$

In summary, we have reduced our original condition for a corrected perturbation, $\text{cossim}(\mathbf{b}'_t(\mathbf{\Delta}) - \mathbf{b}_t, \mathbf{\Delta}) < 0$, into something more tractable to analyze: $(\mathbf{Q}^\top\mathbf{\Delta})^\top\mathbf{V}(\mathbf{Q}^\top\mathbf{\Delta}) < 0$. Observe that $\mathbf{Q}^\top\mathbf{\Delta} \in \mathbb{R}^{d_{\text{model}}}$ is a vector of the perturbation projected into the Jacobian's eigenvector space. In, 2, each projected component is squared and multiplied by the corresponding eigenvalue. Thus, if a perturbation decomposes heavily onto an eigenvector with a negative eigenvalue, our original condition for "correction" will be satisfied. □

More concretely, consider the perturbation from earlier, $\mathbf{a}_t$. We can project $\mathbf{a}_t$ into eigenvector space to obtain $\mathbf{q} = \mathbf{Q}^\top\mathbf{a}_t$. Observe that entry $\mathbf{q}_i$ is the projection onto the i-th eigenvector with eigenvalue λ_i. From the derivation in 2, we have

$$\langle \overline{\mathbf{J}}\mathbf{a}_t, \mathbf{a}_t \rangle = (\mathbf{Q}^\top\mathbf{a}_t)^\top\mathbf{V}(\mathbf{Q}^\top\mathbf{a}_t) = \mathbf{q}^\top\mathbf{V}\mathbf{q} = \sum_i \lambda_i |\mathbf{q}_i|^2 \tag{3}$$

Thus, if $\sum_i \lambda_i |\mathbf{q}_i|^2 < 0$, $\mathbf{a}_t$ is being corrected. Additionally, any individual $\lambda_i |\mathbf{q}|_i^2 < 0$ corresponds to an individual direction that is being corrected, specifically the i-th eigenvector, with correction strength λ_i. Suppose we consider the top eigenvectors, namely those with the top fifty $|\mathbf{q}_i|^2$ values; these fifty directions account for $15{+}\%$ of the variance of $\mathbf{a}_t$. Then, we can plot the distribution of the corresponding λ_i to understand how TLCM interacts with the top components of $\mathbf{a}_t$.

Results. We compute the described plot—the histogram of λ_i for top directions of $\mathbf{a}_t$—across 50 Jacobian-$\mathbf{a}_t$ pairs where TLCM is present and 50 pairs where TLCM is not present. Figure 4b shows the aggregated results. We observe a bimodal eigenvalue distribution for TLCM pairs. The strongest mode hovers near $\lambda = -1$, which means that a unit increase in this direction from the prior layer results in a unit-sized correction.

Importantly, a mode around $\lambda = -1$ implies correction *rather than* partial attenuation of undesired features; the features are entirely reversed. The other mode hovers around $\lambda = 0.5$, which implies that there are directions being *promoted* by TLCM, contrary to our initial expectation. This bimodal distribution is firm evidence that TLCM does not seek to attenuate the previous layer uniformly. In fact, it selects key subspaces to correct, disregarding or promoting the rest. For adjacent layers without TLCM, we observe one strong peak at $\lambda = 0$; this suggests that adjacent layers without TLCM do not interact significantly with the prior layer.

Finally, we observe that about 3000 of the total 4096 eigenvalues of $\overline{\mathbf{J}}$ are negative; see Figure 4a.

5.3 Propose and Reject Hypothesis

We have shown that TLCM does not uniformly reverse the prior layer's contribution but rather selectively corrects a subspace of it. To synthesize our findings, we propose the propose-and-reject hypothesis as a conceptual framework:

Propose-and-reject hypothesis (P&R). TLCM contributes to feature enrichment through a two-stage process: (1) a layer proposes a set of candidate features, and (2) the subsequent layer, equipped with attention mechanisms to gather context, removes irrelevant features.

P&R is consistent with our experiments thus far: Attention and MLP layers both play a role in TLCM; TLCM activates more frequently later in the context window; TLCM activates predominately in the first two-thirds of models, which is connected with enrichment; TLCM corrects *only* the prior layer; TLCM appears to perform selective correction rather than uniform attenuation, as shown in Sec 5.2; among others. We caveat that we do not explicitly test P&R in this work.

P&R could prove particularly valuable for contextual processing. For example, consider the token "202" within "1,808,202". During the forward pass, layer i at this token might propose a handful of feature vectors: "hundred", "thousand", "million". Layer $i+1$—equipped with an attention sublayer—can evaluate and eliminate the incorrect features. The incorrect features would form the "undesirable" subspace which would be corrected by layer $i+1$. The correct feature, "million", would reside in the "desirable" subspace and hence remain untouched.

6 Discussion

Our TLCM experiments have consequences for broader work in mechanistic interpretability. For example, since SAEs interpret residual stream contributions, TLCM's existence predicts *misfires*—features contributed to the residual stream that are immediately reversed by the subsequent layer. In addition, our research also offers a framework for explaining several challenges in SAE-based interpretability:

Feature descriptions lack high specificity. Recent work [47] observed that over 50% of activated features from an SAE are labeled by a large LLM as "Irrelevant" or "Only vaguely related" to the text on which they fire. Highly-activating features, though more rare, tend to show greater specificity to the text. This pattern of low specificity in most features aligns with our expectation for "misfiring" features that are subsequently corrected by the next layer.

Effective model steering requires overcoming TLCM's correction. While we can amplify selected features to steer model behavior, our work predicts that some amplifications will be neutralized by TLCM's corrections. As shown in Appendix E, TLCM's correction capacity begins to diminishes when the prior layer is amplified beyond $2\times$; therefore, feature amplification likely needs to exceed a critical threshold to be effective. Consistent with this hypothesis, recent work demonstrated that effective steering requires extreme amplification levels—up to $10\times$ a feature's maximum observed value [47]. We propose that solving an alternative optimization problem—intervening on feature directions that TLCM is not targeting, identified using the Jacobian—could be a promising direction for future work.

Cross-layer transcoders outperform SAEs. Recent work finds that the semantic meaning of the feature $\mathbf{v}_i$ and $-\mathbf{v}_i$ is unrelated [10]. For example, assume the former means "firetruck" and the latter means "Ancient Greece." If layer i contributes $-\mathbf{v}_i$, an SAE is unable to determine whether layer i aims to correct a faulty "firetruck" activation from the previous layer or contribute a novel "Ancient Greece" feature. Cross-layer transcoders, which are conditioned on the input residual stream, fundamentally can distinguish between these two cases, thus enabling them to learn better features—CLTs can learn both a "firetruck misfire" and an "Ancient Greece" feature that map to the same output vector. Indeed, recent work finds CLTs beat SAEs on certain metrics [40].

More broadly, we think further understanding the architectural causes for TLCM (Appendix A explores some ideas) is exciting subsequent work, as is trying to understand TLCM's corrections in feature extracted via standard interpretability techniques. We hope TLCM helps improve methods to steer LLM's forward passes and is a step towards making models more interpretable and controllable.

References

[1] Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Qin Cai, Vishrav Chaudhary, Dong Chen, Dongdong Chen, Weizhu Chen, Yen-Chun Chen, Yi-Ling Chen, Hao Cheng, Parul Chopra, Xiyang Dai, Matthew Dixon, Ronen Eldan, Victor Fragoso, Jianfeng Gao, Mei Gao, Min Gao, Amit Garg, Allie Del Giorno, Abhishek Goswami, Suriya Gunasekar, Emman Haider, Junheng Hao, Russell J. Hewett, Wenxiang Hu, Jamie Huynh, Dan Iter, Sam Ade Jacobs, Mojan Javaheripi, Xin Jin, Nikos Karampatziakis, Piero Kauffmann, Mahoud Khademi, Dongwoo Kim, Young Jin Kim, Lev Kurilenko, James R. Lee, Yin Tat Lee, Yuanzhi Li, Yunsheng Li, Chen Liang, Lars Liden, Xihui Lin, Zeqi Lin, Ce Liu, Liyuan Liu, Mengchen Liu, Weishung Liu, Xiaodong Liu, Chong Luo, Piyush Madan, Ali Mahmoudzadeh, David Majercak, Matt Mazzola, Caio César Teodoro Mendes, Arindam Mitra, Hardik Modi, Anh Nguyen, Brandon Norick, Barun Patra, Daniel Perez-Becker, Thomas Portet, Reid Pryzant, Heyang Qin, Marko Radmilac, Liliang Ren, Gustavo de Rosa, Corby Rosset, Sambudha Roy, Olatunji Ruwase, Olli Saarikivi, Amin Saied, Adil Salim, Michael Santacroce, Shital Shah, Ning Shang, Hiteshi Sharma, Yelong Shen, Swadheen Shukla, Xia Song, Masahiro Tanaka, Andrea Tupini, Praneetha Vaddamanu, Chunyu Wang, Guanhua Wang, Lijuan Wang, Shuohang Wang, Xin Wang, Yu Wang, Rachel Ward, Wen Wen, Philipp Witte, Haiping Wu, Xiaoxia Wu, Michael Wyatt, Bin Xiao, Can Xu, Jiahang Xu, Weijian Xu, Jilong Xue, Sonali Yadav, Fan Yang, Jianwei Yang, Yifan Yang, Ziyi Yang, Donghan Yu, Lu Yuan, Chenruidong Zhang, Cyril Zhang, Jianwen Zhang, Li Lyna Zhang, Yi Zhang, Yue Zhang, Yunan Zhang, and Xiren Zhou. Phi-3 technical report: A highly capable language model locally on your phone, 2024. URL `https://arxiv.org/abs/2404.14219`.

[2] Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J. Hewett, Mojan Javaheripi, Piero Kauffmann, James R. Lee, Yin Tat Lee, Yuanzhi Li, Weishung Liu, Caio C. T. Mendes, Anh Nguyen, Eric Price, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Xin Wang, Rachel Ward, Yue Wu, Dingli Yu, Cyril Zhang, and Yi Zhang. Phi-4 technical report, 2024. URL `https://arxiv.org/abs/2412.08905`.

[3] Guillaume Alain. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016.

[4] Emmanuel Ameisen, Jack Lindsey, Adam Pearce, Wes Gurnee, Nicholas L. Turner, Brian Chen, Craig Citro, David Abrahams, Shan Carter, Basil Hosmer, Jonathan Marcus, Michael Sklar, Adly Templeton, Trenton Bricken, Callum McDougall, Hoagy Cunningham, Thomas Henighan, Adam Jermyn, Andy Jones, Andrew Persic, Zhenyi Qi, T. Ben Thompson, Sam Zimmerman, Kelley Rivoire, Thomas Conerly, Chris Olah, and Joshua Batson. Circuit tracing: Revealing computational graphs in language models. *Transformer Circuits Thread*, 2025. URL `https://transformer-circuits.pub/2025/attribution-graphs/methods.html`.

[5] Nora Belrose, Zach Furman, Logan Smith, Danny Halawi, Igor Ostrovsky, Lev McKinney, Stella Biderman, and Jacob Steinhardt. Eliciting latent predictions from transformers with the tuned lens, 2023. URL `https://arxiv.org/abs/2303.08112`.

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

[7] Greyson Brothers, Willa Mannering, Amber Tien, and John Winder. Uncovering uncertainty in transformer inference, 2024. URL `https://arxiv.org/abs/2412.05768`.

[8] Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. Discovering latent knowledge in language models without supervision. *arXiv preprint arXiv:2212.03827*, 2022.

[9] Sviatoslav Chalnev, Matthew Siu, and Arthur Conmy. Improving steering vectors by targeting sparse autoencoder features, 2024. URL `https://arxiv.org/abs/2411.02193`.

[10] Dami Choi, Vincent Huang, Kevin Meng, Daniel D Johnson, Jacob Steinhardt, and Sarah Schwettmann. Scaling automatic neuron description. https://transluce.org/neuron-descriptions, October 2024.

[11] Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models, 2023. URL https://arxiv.org/abs/2309.08600.

[12] Jacob Dunefsky, Philippe Chlenski, and Neel Nanda. Transcoders find interpretable llm feature circuits, 2024. URL https://arxiv.org/abs/2406.11944.

[13] Jiahai Feng, Stuart Russell, and Jacob Steinhardt. Monitoring latent world states in language models with propositional probes. *arXiv preprint arXiv:2406.19501*, 2024.

[14] 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, 2024. URL https://arxiv.org/abs/2406.04093.

[15] Atticus Geiger, Zhengxuan Wu, Christopher Potts, Thomas Icard, and Noah Goodman. Finding alignments between interpretable causal variables and distributed neural representations. In *Causal Learning and Reasoning*, pages 160–187. PMLR, 2024.

[16] Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. Dissecting recall of factual associations in auto-regressive language models, 2023. URL https://arxiv.org/abs/2304.14767.

[17] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal

Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3

herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

[18] Dirk Groeneveld, Iz Beltagy, Pete Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Harsh Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, Shane Arora, David Atkinson, Russell Authur, Khyathi Raghavi Chandu, Arman Cohan, Jennifer Dumas, Yanai Elazar, Yuling Gu, Jack Hessel, Tushar Khot, William Merrill, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew E. Peters, Valentina Pyatkin, Abhilasha Ravichander, Dustin Schwenk, Saurabh Shah, Will Smith, Emma Strubell, Nishant Subramani, Mitchell Wortsman, Pradeep Dasigi, Nathan Lambert, Kyle Richardson, Luke Zettlemoyer, Jesse Dodge, Kyle Lo, Luca Soldaini, Noah A. Smith, and Hannaneh Hajishirzi. Olmo: Accelerating the science of language models, 2024. URL https://arxiv.org/abs/2402.00838.

[19] Roee Hendel, Mor Geva, and Amir Globerson. In-context learning creates task vectors, 2023. URL https://arxiv.org/abs/2310.15916.

[20] Evan Hernandez, Belinda Z. Li, and Jacob Andreas. Inspecting and editing knowledge representations in language models, 2024. URL https://arxiv.org/abs/2304.00740.

[21] Evan Hernandez, Arnab Sen Sharma, Tal Haklay, Kevin Meng, Martin Wattenberg, Jacob Andreas, Yonatan Belinkov, and David Bau. Linearity of relation decoding in transformer language models. In *Proceedings of the 2024 International Conference on Learning Representations*, 2024.

[22] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023. URL https://arxiv.org/abs/2310.06825.

[23] Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of experts, 2024. URL https://arxiv.org/abs/2401.04088.

[24] Jiachen Jiang, Jinxin Zhou, and Zhihui Zhu. Tracing representation progression: Analyzing and enhancing layer-wise similarity, 2025. URL https://arxiv.org/abs/2406.14479.

[25] Vedang Lad, Wes Gurnee, and Max Tegmark. The remarkable robustness of llms: Stages of inference? *arXiv preprint arXiv:2406.19384*, 2024.

[26] Belinda Z. Li, Maxwell I. Nye, and Jacob Andreas. Implicit representations of meaning in neural language models. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 1813–1827. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.ACL-LONG.143. URL https://doi.org/10.18653/v1/2021.acl-long.143.

[27] Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model, july 2023. *URL http://arxiv.org/abs/2306.03341*, 2023.

[28] Jack Lindsey, Wes Gurnee, Emmanuel Ameisen, Brian Chen, Adam Pearce, Nicholas L. Turner, Craig Citro, David Abrahams, Shan Carter, Basil Hosmer, Jonathan Marcus, Michael Sklar, Adly Templeton, Trenton Bricken, Callum McDougall, Hoagy Cunningham, Thomas Henighan, Adam Jermyn, Andy Jones, Andrew Persic, Zhenyi Qi, T. Ben Thompson, Sam Zimmerman, Kelley Rivoire, Thomas Conerly, Chris Olah, and Joshua Batson. On the biology of a large language model. *Transformer Circuits Thread*, 2025. URL https://transformer-circuits.pub/2025/attribution-graphs/biology.html.

[29] Alex Mallen and Nora Belrose. Eliciting latent knowledge from quirky language models. *arXiv preprint arXiv:2312.01037*, 2023.

[30] 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, 2025. URL https://arxiv.org/abs/2403.19647.

[31] Callum McDougall, Arthur Conmy, Cody Rushing, Thomas McGrath, and Neel Nanda. Copy suppression: Comprehensively understanding a motif in language model attention heads. In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 337–363, 2024.

[32] Thomas McGrath, Matthew Rahtz, Janos Kramar, Vladimir Mikulik, and Shane Legg. The hydra effect: Emergent self-repair in language model computations, 2023. URL https://arxiv.org/abs/2307.15771.

[33] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. *Advances in Neural Information Processing Systems*, 35:17359–17372, 2022.

[34] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models, 2016.

[35] Tomáš Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: Human language technologies*, pages 746–751, 2013.

[36] Neel Nanda, S Rajamanoharan, J Kramár, and R Shah. Fact finding: Attempting to reverse-engineer factual recall on the neuron level. In *AI Alignment Forum, 2023c. URL https://www. alignmentforum. org/posts/iGuwZTHWb6DFY3sKB/fact-finding-attempting-to-reverse-engineer-factual-recall*, page 19, 2023.

[37] nostalgebraist. interpreting gpt: the logit lens, 2020. URL https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens.

[38] Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. Zoom in: An introduction to circuits. *Distill*, 5(3):e00024–001, 2020.

[39] Alexander Pan, Lijie Chen, and Jacob Steinhardt. Latentqa: Teaching llms to decode activations into natural language. *arXiv preprint arXiv:2412.08686*, 2024.

[40] Gonçalo Paulo, Stepan Shabalin, and Nora Belrose. Transcoders beat sparse autoencoders for interpretability, 2025. URL https://arxiv.org/abs/2501.18823.

[41] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI*, 2019. URL https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf. Accessed: 2024-11-15.

[42] Fabien Roger, Ryan Greenblatt, Max Nadeau, Buck Shlegeris, and Nate Thomas. Benchmarks for detecting measurement tampering, 2023.

[43] Cody Rushing and Neel Nanda. Explorations of self-repair in language models. *arXiv preprint arXiv:2402.15390*, 2024.

[44] Oscar Skean, Md Rifat Arefin, Dan Zhao, Niket Patel, Jalal Naghiyev, Yann LeCun, and Ravid Shwartz-Ziv. Layer by layer: Uncovering hidden representations in language models, 2025. URL https://arxiv.org/abs/2502.02013.

[45] Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Pier Giuseppe Sessa, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, Amélie Héliou, Andrea Tacchetti, Anna Bulanova,

Antonia Paterson, Beth Tsai, Bobak Shahriari, Charline Le Lan, Christopher A. Choquette-Choo, Clément Crepy, Daniel Cer, Daphne Ippolito, David Reid, Elena Buchatskaya, Eric Ni, Eric Noland, Geng Yan, George Tucker, George-Christian Muraru, Grigory Rozhdestvenskiy, Henryk Michalewski, Ian Tenney, Ivan Grishchenko, Jacob Austin, James Keeling, Jane Labanowski, Jean-Baptiste Lespiau, Jeff Stanway, Jenny Brennan, Jeremy Chen, Johan Ferret, Justin Chiu, Justin Mao-Jones, Katherine Lee, Kathy Yu, Katie Millican, Lars Lowe Sjoesund, Lisa Lee, Lucas Dixon, Machel Reid, Maciej Mikuła, Mateo Wirth, Michael Sharman, Nikolai Chinaev, Nithum Thain, Olivier Bachem, Oscar Chang, Oscar Wahltinez, Paige Bailey, Paul Michel, Petko Yotov, Rahma Chaabouni, Ramona Comanescu, Reena Jana, Rohan Anil, Ross McIlroy, Ruibo Liu, Ryan Mullins, Samuel L Smith, Sebastian Borgeaud, Sertan Girgin, Sholto Douglas, Shree Pandya, Siamak Shakeri, Soham De, Ted Klimenko, Tom Hennigan, Vlad Feinberg, Wojciech Stokowiec, Yu hui Chen, Zafarali Ahmed, Zhitao Gong, Tris Warkentin, Ludovic Peran, Minh Giang, Clément Farabet, Oriol Vinyals, Jeff Dean, Koray Kavukcuoglu, Demis Hassabis, Zoubin Ghahramani, Douglas Eck, Joelle Barral, Fernando Pereira, Eli Collins, Armand Joulin, Noah Fiedel, Evan Senter, Alek Andreev, and Kathleen Kenealy. Gemma: Open models based on gemini research and technology, 2024. URL `https://arxiv.org/abs/2403.08295`.

[46] Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, Anton Tsitsulin, Nino Vieillard, Piotr Stanczyk, Sertan Girgin, Nikola Momchev, Matt Hoffman, Shantanu Thakoor, Jean-Bastien Grill, Behnam Neyshabur, Olivier Bachem, Alanna Walton, Aliaksei Severyn, Alicia Parrish, Aliya Ahmad, Allen Hutchison, Alvin Abdagic, Amanda Carl, Amy Shen, Andy Brock, Andy Coenen, Anthony Laforge, Antonia Paterson, Ben Bastian, Bilal Piot, Bo Wu, Brandon Royal, Charlie Chen, Chintu Kumar, Chris Perry, Chris Welty, Christopher A. Choquette-Choo, Danila Sinopalnikov, David Weinberger, Dimple Vijaykumar, Dominika Rogozińska, Dustin Herbison, Elisa Bandy, Emma Wang, Eric Noland, Erica Moreira, Evan Senter, Evgenii Eltyshev, Francesco Visin, Gabriel Rasskin, Gary Wei, Glenn Cameron, Gus Martins, Hadi Hashemi, Hanna Klimczak-Plucińska, Harleen Batra, Harsh Dhand, Ivan Nardini, Jacinda Mein, Jack Zhou, James Svensson, Jeff Stanway, Jetha Chan, Jin Peng Zhou, Joana Carrasqueira, Joana Iljazi, Jocelyn Becker, Joe Fernandez, Joost van Amersfoort, Josh Gordon, Josh Lipschultz, Josh Newlan, Ju yeong Ji, Kareem Mohamed, Kartikeya Badola, Kat Black, Katie Millican, Keelin McDonell, Kelvin Nguyen, Kiranbir Sodhia, Kish Greene, Lars Lowe Sjoesund, Lauren Usui, Laurent Sifre, Lena Heuermann, Leticia Lago, Lilly McNealus, Livio Baldini Soares, Logan Kilpatrick, Lucas Dixon, Luciano Martins, Machel Reid, Manvinder Singh, Mark Iverson, Martin Görner, Mat Velloso, Mateo Wirth, Matt Davidow, Matt Miller, Matthew Rahtz, Matthew Watson, Meg Risdal, Mehran Kazemi, Michael Moynihan, Ming Zhang, Minsuk Kahng, Minwoo Park, Mofi Rahman, Mohit Khatwani, Natalie Dao, Nenshad Bardoliwalla, Nesh Devanathan, Neta Dumai, Nilay Chauhan, Oscar Wahltinez, Pankil Botarda, Parker Barnes, Paul Barham, Paul Michel, Pengchong Jin, Petko Georgiev, Phil Culliton, Pradeep Kuppala, Ramona Comanescu, Ramona Merhej, Reena Jana, Reza Ardeshir Rokni, Rishabh Agarwal, Ryan Mullins, Samaneh Saadat, Sara Mc Carthy, Sarah Cogan, Sarah Perrin, Sébastien M. R. Arnold, Sebastian Krause, Shengyang Dai, Shruti Garg, Shruti Sheth, Sue Ronstrom, Susan Chan, Timothy Jordan, Ting Yu, Tom Eccles, Tom Hennigan, Tomas Kocisky, Tulsee Doshi, Vihan Jain, Vikas Yadav, Vilobh Meshram, Vishal Dharmadhikari, Warren Barkley, Wei Wei, Wenming Ye, Woohyun Han, Woosuk Kwon, Xiang Xu, Zhe Shen, Zhitao Gong, Zichuan Wei, Victor Cotruta, Phoebe Kirk, Anand Rao, Minh Giang, Ludovic Peran, Tris Warkentin, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, D. Sculley, Jeanine Banks, Anca Dragan, Slav Petrov, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Sebastian Borgeaud, Noah Fiedel, Armand Joulin, Kathleen Kenealy, Robert Dadashi, and Alek Andreev. Gemma 2: Improving open language models at a practical size, 2024. URL `https://arxiv.org/abs/2408.00118`.

[47] Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L Turner, Callum McDougall, Monte MacDiarmid, C. Daniel Freeman, Theodore R. Sumers, Edward Rees, Joshua Batson, Adam Jermyn, Shan Carter, Chris Olah, and Tom Henighan. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet.

Transformer Circuits Thread, 2024. URL https://transformer-circuits.pub/2024/scaling-monosemanticity/index.html.

[48] Eric Todd, Millicent L. Li, Arnab Sen Sharma, Aaron Mueller, Byron C. Wallace, and David Bau. Function vectors in large language models. In *Proceedings of the 2024 International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=AwyxtyMwaG. arXiv:2310.15213.

[49] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.emnlp-demos.6.

[50] Zhengxuan Wu, Aryaman Arora, Zheng Wang, Atticus Geiger, Dan Jurafsky, Christopher D Manning, and Christopher Potts. Reft: Representation finetuning for language models. *arXiv preprint arXiv:2404.03592*, 2024.

[51] Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. *arXiv preprint arXiv:2309.17453*, 2023.

[52] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. Qwen2 technical report, 2024. URL https://arxiv.org/abs/2407.10671.

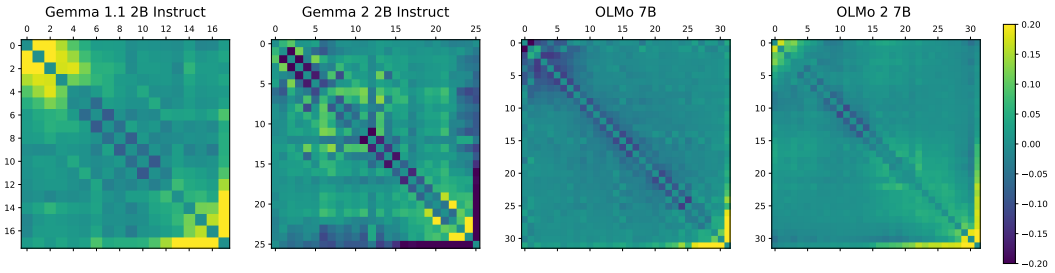


Figure A1: TLCM is consistently found on models with different approaches to LayerNorm. Surprisingly, Gemma 2 with post-LayerNorm has more pronounced TLCM. OLMo 2, which replaces pre-LayerNorm entirely exhibits TLCM. For visual clarity, we plot $\mathrm{clamp}(\mathbf{M}, -0.2, 0.2)$ and zero the diagonals.

A LayerNorm Blindness to explain TLCM

One natural cause of TLCM is RMSNorm. RMSNorm normalizes the input to the attention and MLP sublayers in nearly all of the open-source models analyzed. Formally, for an input $\mathbf{x} \in \mathbb{R}^{d_\mathrm{m}}$ to an attention or MLP sublayers and a learnable parameter vector $\mathbf{g}$, RMSNorm is defined as:

$$\overline{\mathbf{x}} = \frac{\mathbf{x}}{\mathrm{RMS}(\mathbf{x})} \odot \mathbf{g} \qquad \mathrm{RMS}(\mathbf{x}) = \frac{1}{\sqrt{d_\mathrm{m}}} \|\mathbf{x}\|_2,$$

where d_m is the dimension of the residual stream.

Crucially, the use of RMSNorm implies that both the attention and MLP sublayers have *layernorm blindness*; they are blind to the norm of the residual stream. This blindness is significant because these sublayers predict contributions to the residual stream (i.e. $\mathbf{x}_i + \mathrm{Sublayer}(\mathbf{x}_i)$), whose relative impacts depend on the the residual streams current magnitude. Without visibility into the residual stream norm, attention and MLP sublayers risk under-contributing when the norm is high, which potentially leads to their contributions being overshadowed. This could encourage over-contribution behaviors followed by correction, which is consistent with TLCM.

However, we find that *LayerNorm blindness alone does not fully explain the TLCM*. This is because as discussed Sec. 5.2, the correction mechanism does not entirely reverse $\mathbf{a}_t$, contrary to what would be expected if RMSNorm were the primary cause. But perhaps more critically, models trained with alternative RMSNorm implementations continue to exhibit the mechanism:

Gemma. The sublayers in Gemma 1 were trained using pre-LayerNorm as described above. In contrast, Gemma 2 utilized a hybrid approach, employing both pre-LayerNorm and post-LayerNorm:

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \mathrm{LayerNorm}(\mathrm{Sublayer}(\mathrm{LayerNorm}(\mathbf{x}_i))).$$

The addition of post-LayerNorm should, in principle, make the residual stream norm more predictable. However, empirical results show that the correction mechanism remains robust in Gemma 2. Refer to Figure A1 for a comparison.

OLMo. The original OLMo models employed pre-LayerNorm exclusively. In the OLMo 2 series, pre-LayerNorm was replaced with post-LayerNorm and QK norm. Despite this architectural change, the correction mechanism persists strongly in OLMo 2, as shown in Figure A1.

B Likelihood of Negative Cosine Similarity

B.1 Random Normal IID Vectors

Consider two random vectors: $\mathbf{u}, \mathbf{v} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_d)$.

$$
\begin{aligned}
\mathbb{E}[\text{cossim}(\mathbf{u}, \mathbf{v})] &= \mathbb{E}\left[\frac{\langle \mathbf{u}, \mathbf{v} \rangle}{\|\mathbf{u}\|_2 \|\mathbf{v}\|_2}\right] \\
&= \mathbb{E}\left[\frac{\sum_i \mathbf{u}_i \mathbf{v}_i}{\sqrt{\sum_i \mathbf{u}_i^2}\sqrt{\sum_i \mathbf{v}_i^2}}\right] \\
&= \mathbb{E}\left[\sum_j \frac{\mathbf{u}_j}{\sqrt{\sum_i \mathbf{u}_i^2}} \frac{\mathbf{v}_j}{\sqrt{\sum_i \mathbf{v}_i^2}}\right] \\
&= \sum_j \mathbb{E}\left[\frac{\mathbf{u}_j}{\sqrt{\sum_i \mathbf{u}_i^2}}\right] \mathbb{E}\left[\frac{\mathbf{v}_j}{\sqrt{\sum_i \mathbf{v}_i^2}}\right] \\
&= 0
\end{aligned}
$$

The last expectations must be 0 because $\frac{\mathbf{u}_j}{\sqrt{\sum_i \mathbf{u}_i^2}}$ is distributed the same as $\frac{-\mathbf{u}_j}{\sqrt{\sum_i \mathbf{u}_i^2}}$. To calculate the variance, first observe the following fact:

$$
\begin{aligned}
\mathbb{E}\left[\sum_{j=1}^{d} \frac{\mathbf{u}_j^2}{\sum_i \mathbf{u}_i^2}\right] &= 1 \\
\sum_{j=1}^{d} \mathbb{E}\left[\frac{\mathbf{u}_j^2}{\sum_i \mathbf{u}_i^2}\right] &= 1 \\
d\mathbb{E}\left[\frac{\mathbf{u}_1^2}{\sum_i \mathbf{u}_i^2}\right] &= 1 \\
\mathbb{E}\left[\frac{\mathbf{u}_1^2}{\sum_i \mathbf{u}_i^2}\right] &= \frac{1}{d}
\end{aligned}
$$

Using this identity, the variance is:

$$
\begin{aligned}
\text{Var}[\text{cossim}(\mathbf{u}, \mathbf{v})] &= \mathbb{E}[\text{cossim}(\mathbf{u}, \mathbf{v})^2] \\
&= \mathbb{E}\left[\frac{(\sum_i \mathbf{u}_i \mathbf{v}_i)(\sum_i \mathbf{u}_i \mathbf{v}_i)}{\sum_i \mathbf{u}_i^2 \sum_i \mathbf{v}_i^2}\right] \\
&= \mathbb{E}\left[\frac{\sum_{i,j} \mathbf{u}_i \mathbf{v}_i \mathbf{u}_j \mathbf{v}_j}{\sum_i \mathbf{u}_i^2 \sum_i \mathbf{v}_i^2}\right] \\
&= \mathbb{E}\left[\frac{\sum_i \mathbf{u}_i^2 \mathbf{v}_i^2}{\sum_i \mathbf{u}_i^2 \sum_i \mathbf{v}_i^2}\right] \\
&= \sum_{j=1}^{d} \mathbb{E}\left[\frac{\mathbf{u}_j^2}{\sum_i \mathbf{u}_i^2}\right] \mathbb{E}\left[\frac{\mathbf{v}_j^2}{\sum_i \mathbf{v}_i^2}\right] \\
&= \sum_{j=1}^{d} \frac{1}{d} \cdot \frac{1}{d} = \frac{1}{d}
\end{aligned}
$$

B.2 Experimental Mean and Standard Deviation

We find that the contributions of a particular layer are anisotropic; they cluster around approximately 500-800 of the 4096 dimensions of Llama's residual stream.

Specifically, we isolate the contribution vector for layer i across 4096 tokens from wikitext: $\{\mathbf{c}_{i,t_1}, \mathbf{c}_{i,t_2}, \ldots, \mathbf{c}_{i,t_{4096}}\}$. After computing the SVD of each set, we plot percent variance explained by top n principal components vs. n in Figure A2.

Due to this anisotropy, we calculate the statistical significance of the TLCM cutoff (-0.1 cosine similarity) empirically. We compute the mean and variance of cosine similarity across contributions of adjacent layers on *different* tokens. More formally, we compute the mean and variance of $\text{cossim}(\mathbf{c}_{i,t_1}, \mathbf{c}_{i+1,t_2})$ for $t_1 \neq t_2$ and $4 \leq i < 20$. We find it has has mean -0.00375 and standard deviation is 0.03267, meaning that our cutoff of -0.1 is conservatively a 3σ event; most TLCM events occur at lower cosine similarities (-0.15 to -0.25).

We plot these distributions by layer in Figure A3.

C Details on Jacobian Experiments

C.0.1 Jacobian Sanity Check

We have approximated our response function $\mathbf{b}'_t$ using the Taylor expansion.

$$\mathbf{b}'_t(\boldsymbol{\Delta}) = \mathbf{b}'_t(0) + \mathbf{J}\boldsymbol{\Delta} + \mathcal{O}(\boldsymbol{\Delta}^2)$$

Here we aim to confirm that the Jacobian is appropriately representative of the transformer layer within a reasonable regime. Observe that the error of this approximation is $\mathbf{b}'_t(\boldsymbol{\Delta}) - \mathbf{b}'_t(0) - \mathbf{J}\boldsymbol{\Delta}$, and we thus denote the percent error of the approximation as follows:

$$\text{err}(\boldsymbol{\Delta}) = \frac{\|\mathbf{b}'_t(\boldsymbol{\Delta}) - \mathbf{b}'_t(0) - \mathbf{J}\boldsymbol{\Delta}\|}{\|\mathbf{b}'_t(\boldsymbol{\Delta}) - \mathbf{b}'_t(0)\|}$$

We plot the percent error error of this approximation across 46 randomly selected TLCM Jacobians for different values of $\boldsymbol{\Delta} = \alpha \mathbf{a}_t$, $\alpha \in \{-0.5, -0.4, -0.3, \ldots, 0.3, 0.4, 0.5\}$. Shown in Figure A4, we find that the Jacobian is a good approximation within this regime. For $|\alpha| < 0.1$, there is consistently around 5% error, which we believe is sufficient for our Jacobian-based analysis in Sec. 5.2.

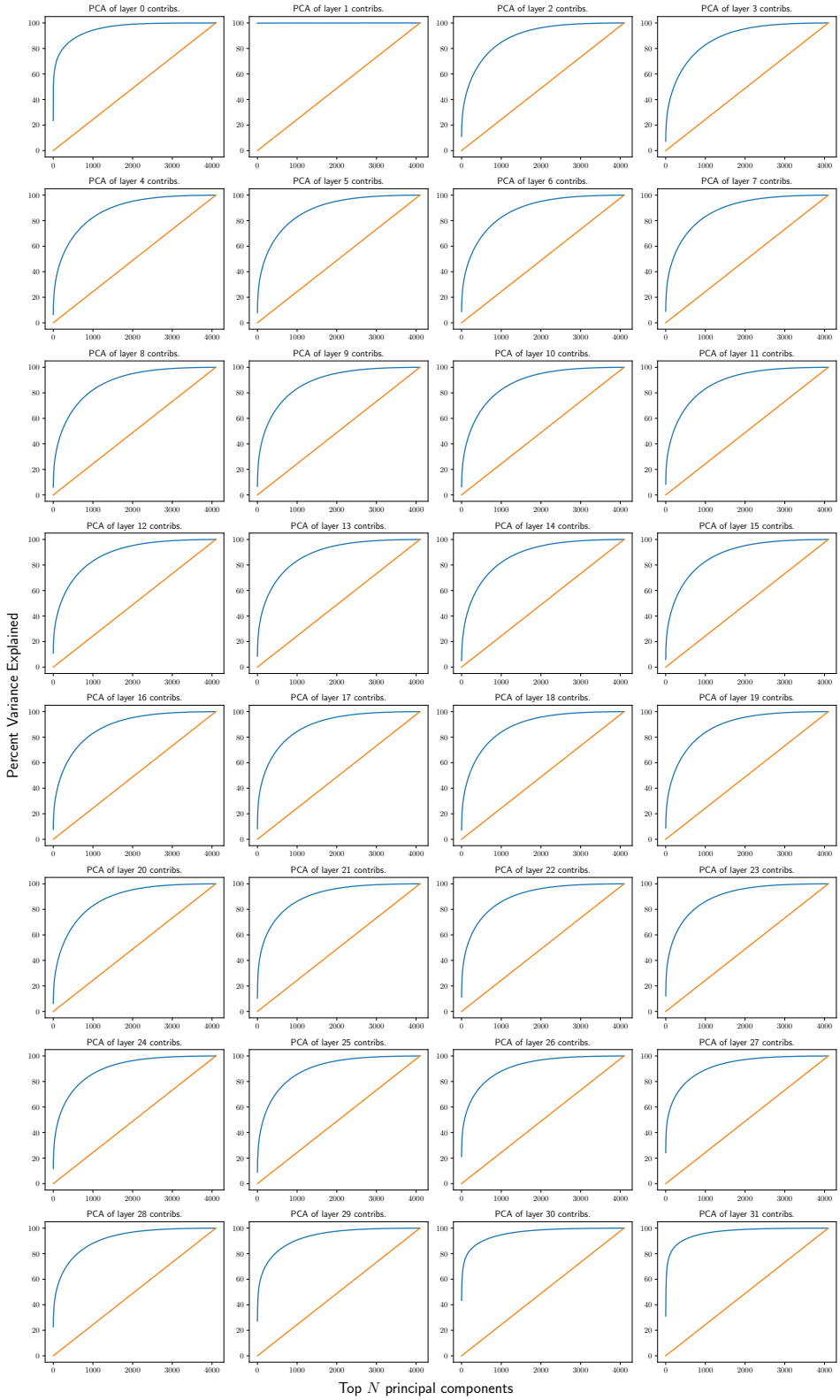


Figure A2: Percent variance explained by the top n principal components vs n, graphed on the blue line. The orange line is what we would expect if contributions were isotropic. For layers where TLCM is most active (4 to 20), the first 500-800 principal components explain 80% of the variance in contributions, demonstrating that contributions are anisotropic.

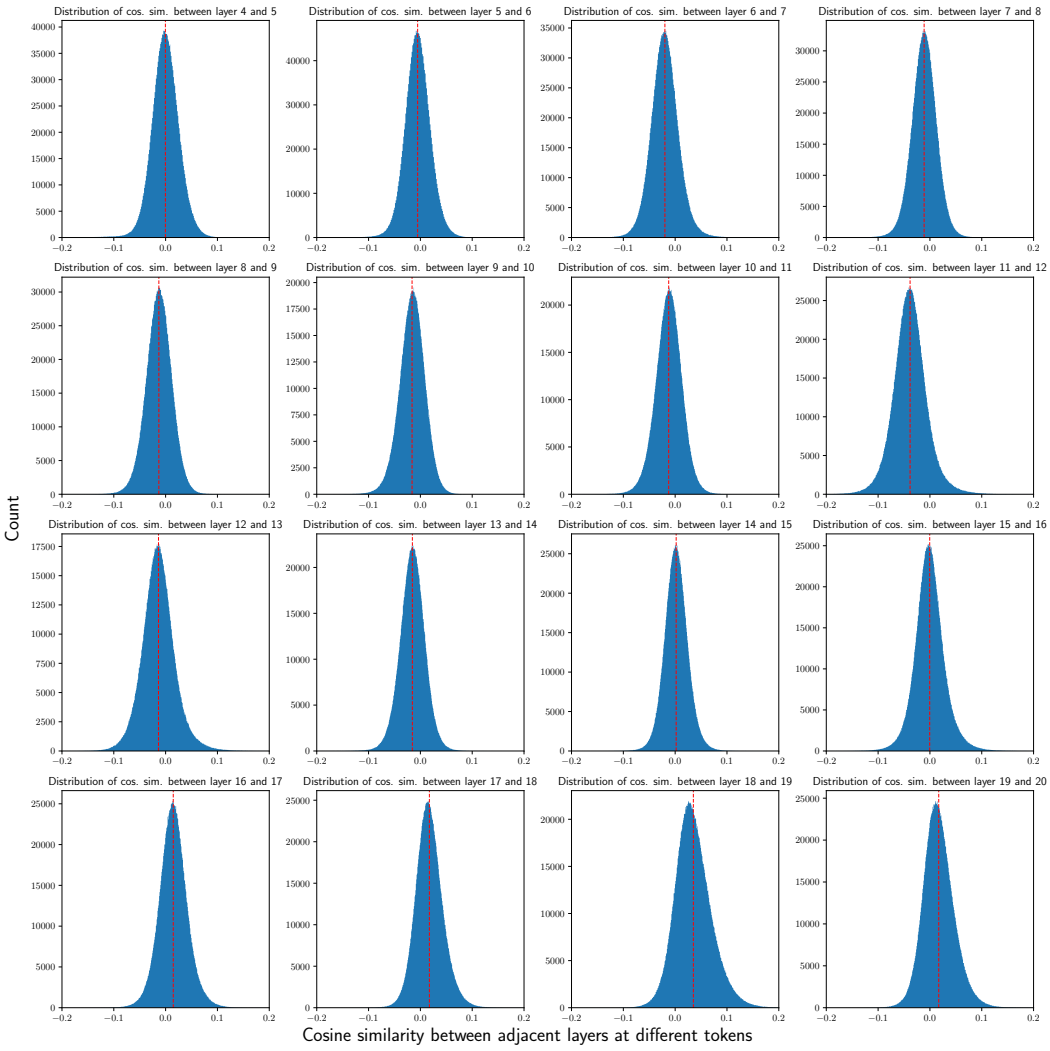


Figure A3: The empirical distribution of $\mathrm{cossim}(\mathbf{c}_{i,t_1}, \mathbf{c}_{i+1,t_2}), t_1 \neq t_2$ across 2000 tokens for layers $4 \leq i < 20$, corresponding to approximately 2 million samples per histogram. The red dotted line corresponds to the mean.

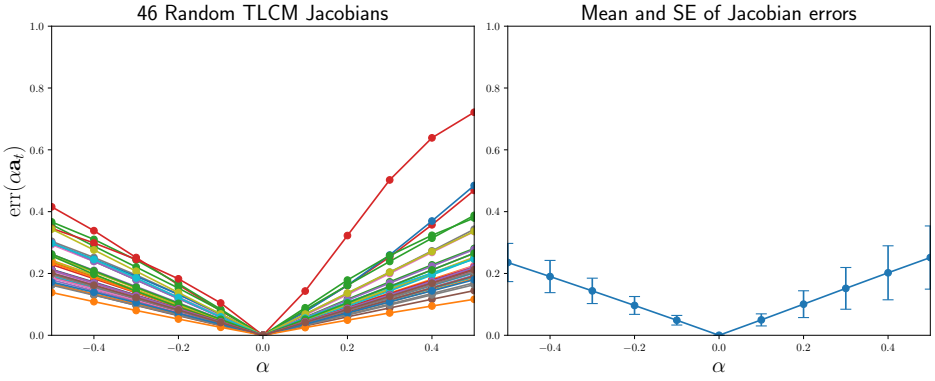


Figure A4: The transformer layer Jacobian is a very good approximation for reasonably large perturbations ($|\alpha| < 0.1$) to the following layer, making it useful for decomposing directions as described in Sec. 5.2.

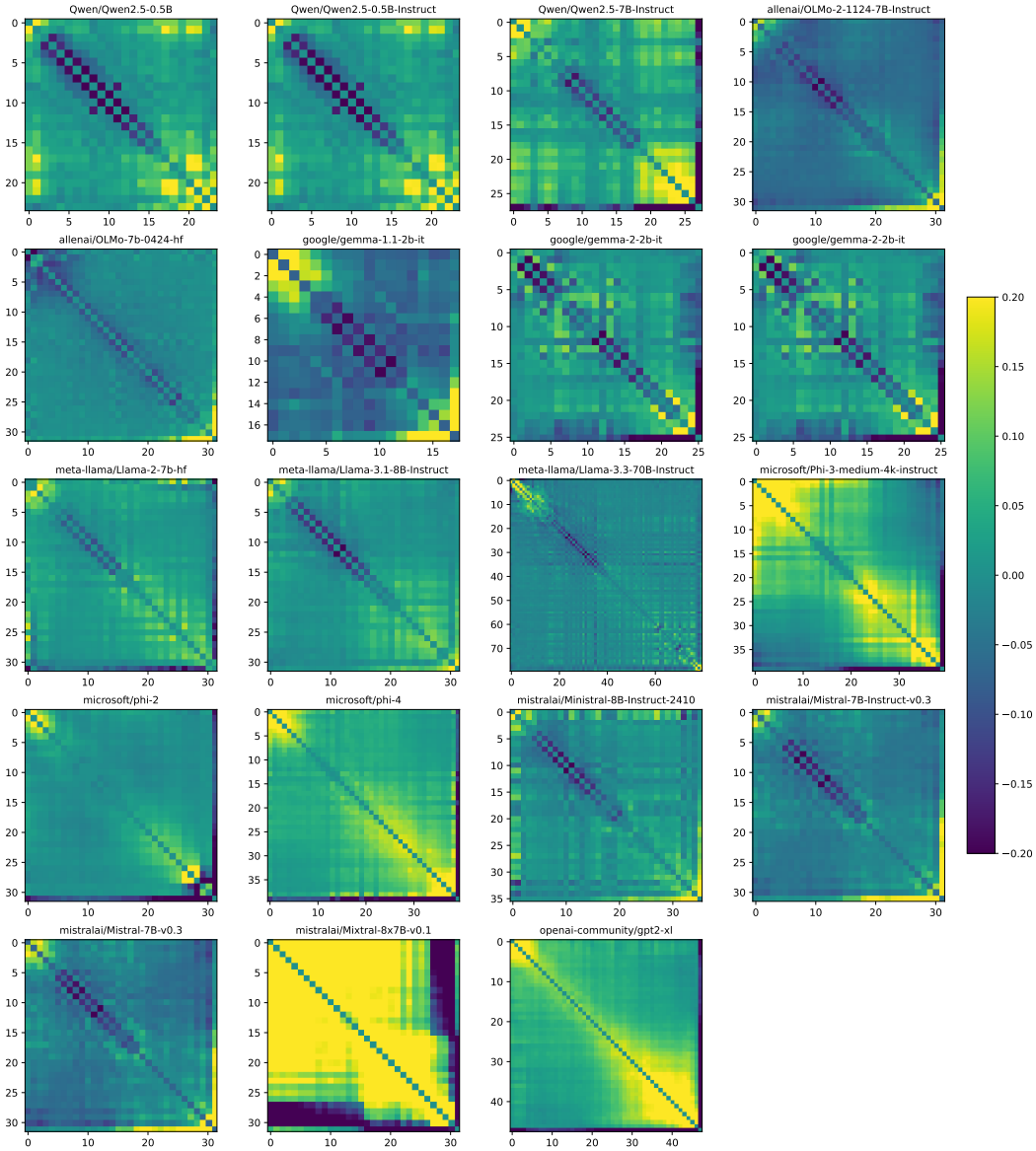


Figure A5: We plot $\mathrm{clamp}(\mathbf{M}, -0.2, 0.2)$ for four models, zeroing diagonal entries. We computed $\mathbf{M}$ across a variety of models, pulled from Huggingface Transformers.

D Extended Details on TLCM Existence

In Figure A5, we plot TLCM's existence across many HuggingFace models using the same technique as described in Sec. 4.1 of the main body.

We previously demonstrated that MLPs exhibit anti-correlations with each other, while attentions do not. We additionally find that MLPs correct attentions (both within and between transformer layers) and that attentions correct MLPs from prior layers. This could be due to a common low-dimensional subspace used by both units for communication. Thus, MLPs correct both prior attention and MLPs; attentions correct just prior MLPs. Altogether, this suggests that MLPs are more responsible for TLCM's correction, but both units are involved.

Specifically, we plot $\mathbf{M}_{\text{attn}\times\text{MLP}} = \frac{1}{n}\sum_t \mathbf{M}_{\text{attn}\times\text{MLP},t}$ where

$$\mathbf{M}_{\text{attn}\times\text{MLP},t}[i,j] := \text{cossim}(\text{Attn}_i(\mathbf{x}_{i,t}), \text{MLP}_j(\mathbf{x}_{j,t}))$$

See Figure A6.

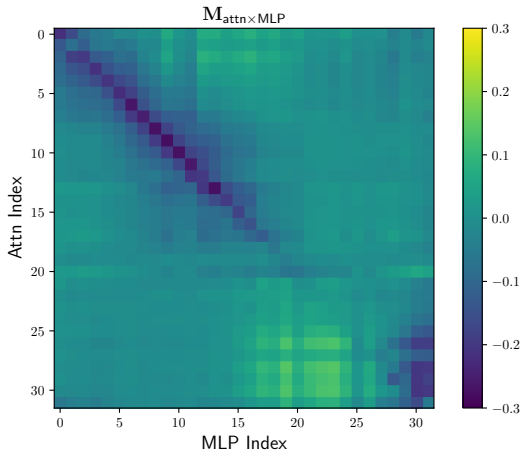


Figure A6: MLPs correct prior attentions, and attentions correct prior MLPs.

E Extended Details on TLCM Adaptivity

E.1 TLCM is Adaptive Across Layers

In Figure A7, we plot TLCM's adaptive correction across different layers using the same technique as described in Sec. 4.1.

E.2 TLCM Correction Capacity Diminishes at High α

In Sec 5.1, we find that TLCM increases its correction as the prior layer is scaled. However, we find that as we scale the prior layer to extremely large values ($\alpha > 2$), the correction capacity of TLCM starts to diminish. Intuitively, as α is scaled, what TLCM previously considered a "mistake" now might be assumed "correct." This could explain why model steering interventions—in which a chosen feature vector is manually contributed to the residual stream—requires features contributions at $10\times$ the maximum ever observed value. In other words, steering interventions must overcome TLCM. See Figure A8 for plots of the correction beginning to diminish.

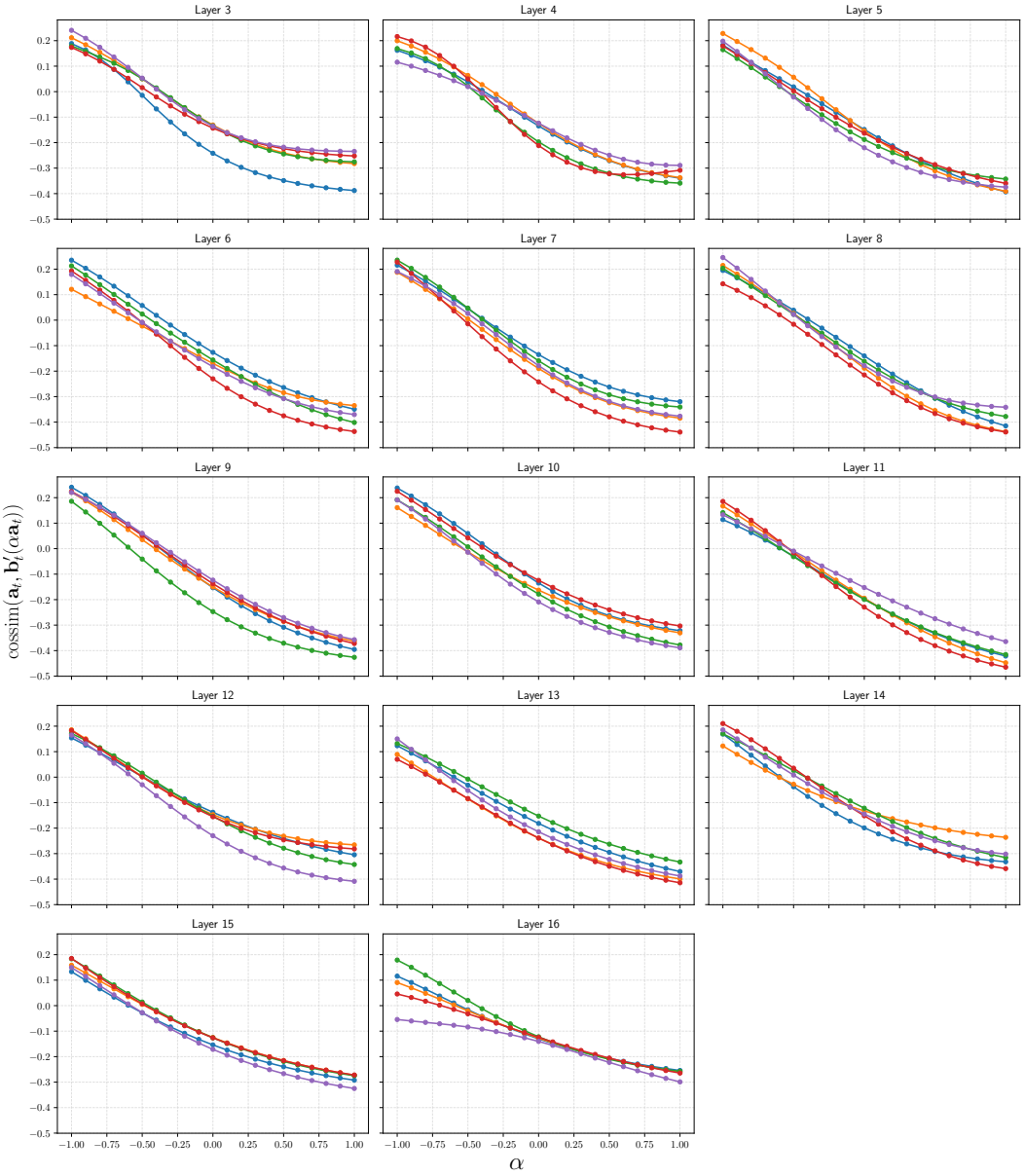


Figure A7: TLCM adaptively regulates the previous layer in a near *linear* fashion. We plot 5 random corrections across a variety of layers in Llama 3.1 8B.

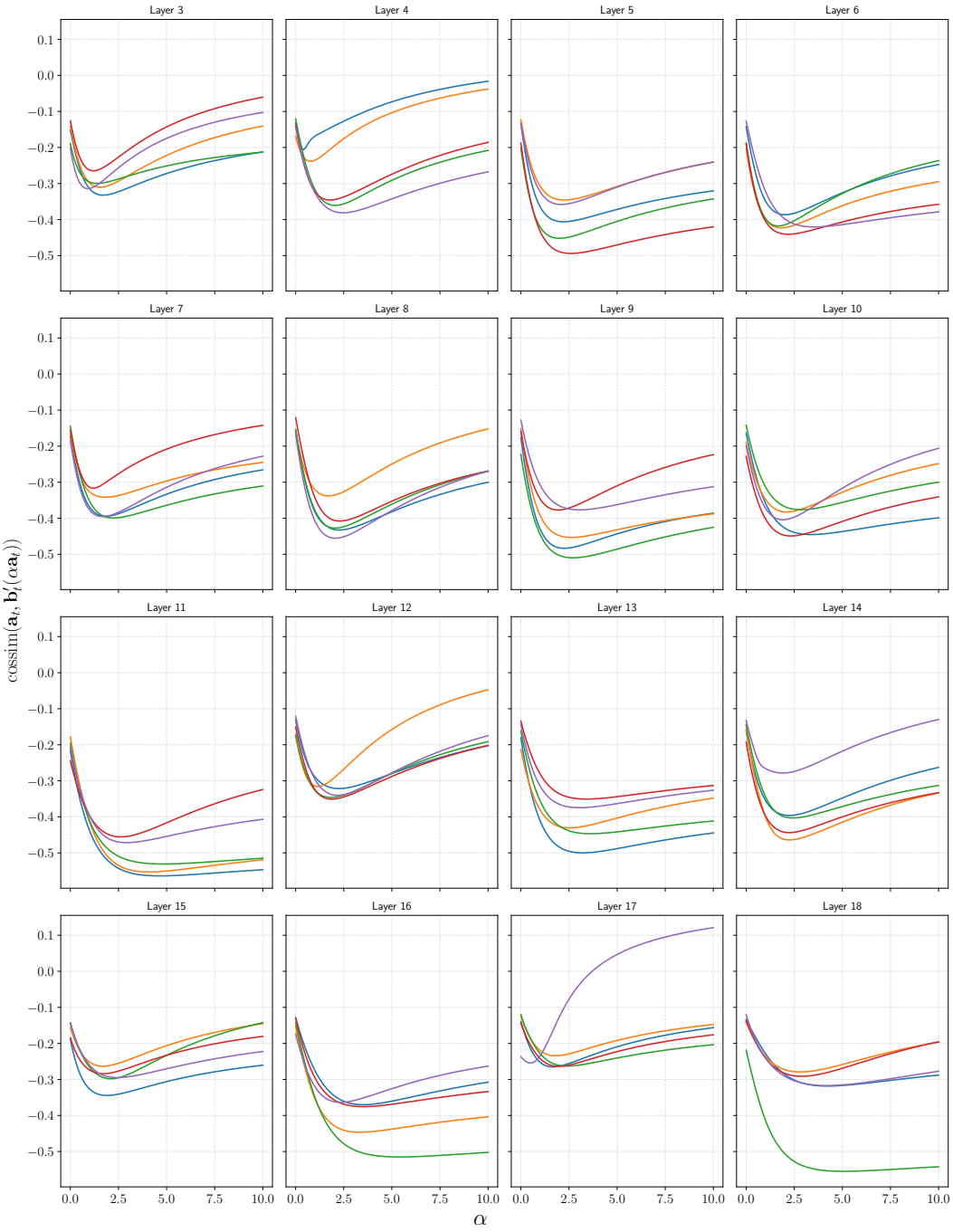


Figure A8: As we increase the previous layer more dramatically, we see TLCM's correction begin to diminish, aside from some outliers. For each layer, we sample 5 random TLCM curves and compute at α increments of 0.2.

F OLMo Training Checkpoint Experiment Details

On a corpus of about 500 tokens of PyTorch instructional content, we compute $\mathbf{M}$ on a handful of training checkpoints, shown in Figure A9.

The corpus is below; beyond a high enough number of tokens, we find this to have little effect on the cosine similarity matrices and thus also the figures.

```
**Using Convolutional Layers in PyTorch**
==================================================

Convolutional layers are a fundamental component of convolutional neural
    networks (CNNs) used for image classification, object detection, and
    other computer vision tasks. In PyTorch, convolutional layers are
    implemented using the ‘nn.Conv2d‘ module.

**Creating a Convolutional Layer**
------------------------------

To create a convolutional layer in PyTorch, you can use the following code:

‘‘‘python
import torch
import torch.nn as nn

# Define the convolutional layer
conv_layer = nn.Conv2d(in_channels, out_channels, kernel_size, stride, padding
    )
‘‘‘

*   ‘in_channels‘: The number of input channels (e.g., 3 for RGB images).
*   ‘out_channels‘: The number of output channels (e.g., 64 for a feature map).

*   ‘kernel_size‘: The size of the convolutional kernel (e.g., 3x3).
*   ‘stride‘: The stride of the convolutional kernel (e.g., 1).
*   ‘padding‘: The amount of padding to apply (e.g., 1).

**Example Usage**
-----------------

Here’s an example of using a convolutional layer in a PyTorch model:

‘‘‘python
import torch
import torch.nn as nn

class ConvNet(nn.Module):
    def __init__(self):
        super(ConvNet, self).__init__()
        self.conv_layer = nn.Conv2d(3, 64, kernel_size=3, stride=1, padding=1)

    def forward(self, x):
        return torch.relu(self.conv_layer(x))

# Initialize the model and input tensor
model = ConvNet()
input_tensor = torch.randn(1, 3, 224, 224)

# Forward pass
output = model(input_tensor)
```

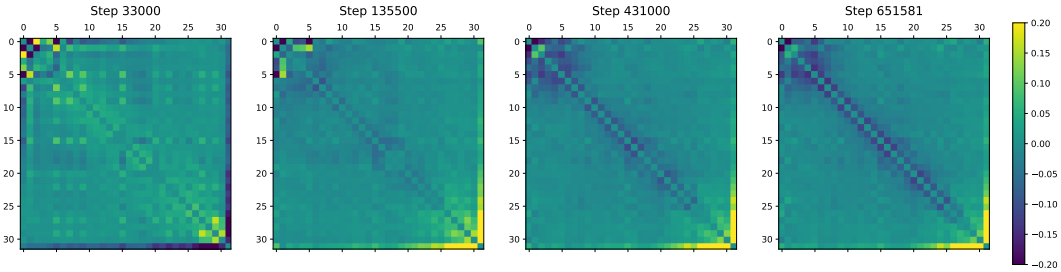


Figure A9: In OLMo 7B, TLCM emerges progressively during pretraining, with initial manifestation around step 135,500 (0.5T tokens). These four plots show $\mathbf{M}$ computed at different training checkpoints of OLMo 1B, with the rightmost plot representing the fully pretrained model. For visual clarity, we plot $\mathrm{clamp}(\mathbf{M}, -0.2, 0.2)$ and zero the diagonals.

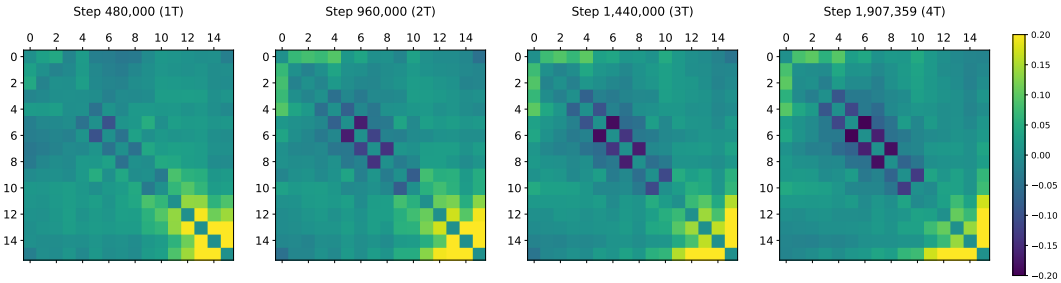


Figure A10: In OLMo 2 1B, TLCM also emerges progressively during pretraining. We plot figures at different step numbers and number of tokens (1, 2, 3, or 4 trillion tokens).

G Token-Level Correction Statistics

G.1 Experiment Prompts

We use the following prompts to generate data for our experiment in Sec 4.3:

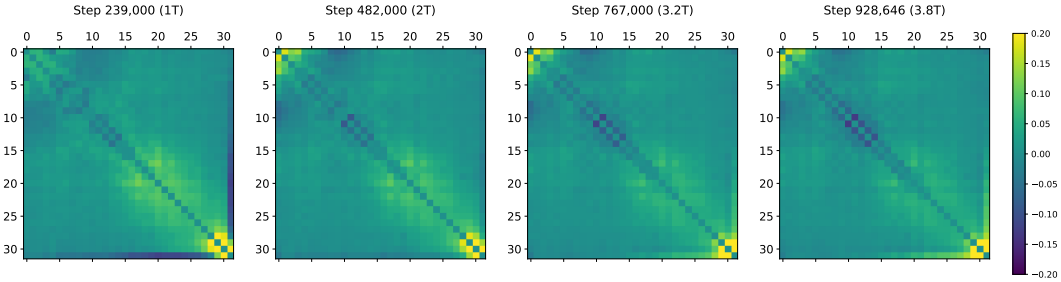


Figure A11: In OLMo 2 7B, TLCM also emerges progressively during pretraining. We plot figures at different step numbers and number of tokens.

Write a blog post about the impact of remote work on urban real estate trends.
Write an essay on the psychological effects of social media on teenagers.
Write a report detailing the advancements in renewable energy technologies over the last decade.
Write an article about the rise of plant-based diets and their environmental benefits.
Write a memo to employees explaining the new company policy on cybersecurity measures.
Write a letter to a local council advocating for improved recycling facilities in the community.
Write a proposal for implementing a mindfulness program in elementary schools to enhance student well-being.
Write a blog post about the evolution of smart home technology and its implications for privacy.
Write an essay discussing the ethical considerations of genetic editing technologies.
Write a report on the economic impacts of the COVID-19 pandemic on small businesses.
Write an article about the significance of the James Webb Space Telescope's latest findings.
Write a memo outlining the steps for a successful digital transformation in a manufacturing company.
Write a letter to a senator expressing concerns about the proposed changes to healthcare laws.
Write a proposal for a community garden project to promote local food production and community engagement.
Write a blog post about the latest trends in artificial intelligence and machine learning.
Write an essay on the role of art therapy in mental health recovery.
Write a report assessing the potential of hydrogen fuel as an alternative energy source.
Write an article highlighting the importance of biodiversity conservation in combating climate change.
Write a memo to staff regarding the integration of a new project management software.
Write a letter to an editor expressing opinions on the local government's transportation plan.
Write a proposal for a telemedicine service to increase healthcare access in rural areas.
Write a blog post discussing the future of space tourism and its possible timeline.
Write an essay exploring the cultural significance of indigenous music.
Write a report on the trends in global unemployment rates and their implications for economic policy.
Write an article about the benefits and challenges of homeschooling.
Write a memo describing the company's strategy to address the upcoming industry regulations.
Write a letter to a non-profit organization offering to partner on an environmental initiative.
Write a proposal for an employee wellness program that includes both physical and mental health activities.
Write a blog post analyzing the impact of blockchain technology on financial services.
Write an essay on the historical impact of major pandemics on societal structures.
Write a report on the viability of vertical farming in urban environments.
Write an article about the challenges of maintaining data privacy in the age of IoT.
Write a memo to update company leadership on the progress of the quarterly goals.
Write a letter to a school board proposing the introduction of coding classes in middle schools.
Write a proposal for a local government initiative to support small businesses during economic downturns.
Write a blog post about the techniques and benefits of sustainable agriculture.
Write an essay on the influence of classical music on modern genres.
Write a report on consumer behavior changes in the automotive industry towards electric vehicles.
Write an article about the role of youth activism in shaping public policy.

Continued on the next page.

Write a memo detailing guidelines for handling customer data under new privacy laws.
Write a letter to the editor about the importance of public parks and open spaces.
Write a proposal for a new arts festival aiming to showcase local and international talent.
Write a blog post on the role of robotics in healthcare and potential ethical dilemmas.
Write an essay about the impact of climate change on marine ecosystems.
Write a report on strategies for managing workplace diversity in a global company.
Write an article on the resurgence of interest in vinyl records and analog music.
Write a memo to department heads about managing remote teams effectively.
Write a letter to a city planner regarding the need for improved pedestrian pathways.
Write a proposal for implementing a bike-sharing program in a mid-sized city.
Write a blog post about the future trends in education technology and their implications for learning.
Write a blog post about the growing popularity of mindfulness apps and their effectiveness.
Write an essay on the resurgence of traditional farming techniques in modern agriculture.
Write a report on the adoption of electric vehicles in major cities around the world.
Write an article about the psychological benefits of outdoor activities.
Write a memo to management detailing the steps to achieve carbon neutrality in the workplace by 2030.
Write a letter to a philanthropic organization requesting funding for a community tech hub.
Write a proposal for a series of workshops aimed at teaching digital literacy to seniors.
Write a blog post analyzing the impact of virtual reality on entertainment and media.
Write an essay discussing the philosophical implications of artificial intelligence surpassing human intelligence.
Write a report on the state of child nutrition programs in public schools.
Write an article about the role of drones in modern agriculture and their environmental impact.
Write a memo regarding the implementation of a flexible work schedule to enhance employee productivity.
Write a letter to a government official advocating for stricter air pollution regulations.
Write a proposal for a new public library with advanced digital resources.
Write a blog post about the importance of cybersecurity in the age of cloud computing.
Write an essay exploring the historical role of spices in global trade.
Write a report on the effectiveness of recent public health campaigns on smoking cessation.
Write an article on the growing trend of micro-living and tiny homes.
Write a memo introducing a new internal team dedicated to innovation and strategic initiatives.
Write a letter to parents outlining the new curriculum changes in a local school district.
Write a proposal for a mobile health clinic to serve underserved areas.
Write a blog post about the use of big data in personalized medicine.
Write an essay on the evolution of language in the digital age.
Write a report detailing the economic impact of cultural festivals on local communities.
Write an article on the significance of urban green spaces for mental health.
Write a memo to staff about upcoming training opportunities in advanced analytics.
Write a letter to the editor discussing the need for more inclusive sports programs in schools.
Write a proposal for an annual technology conference focusing on sustainability innovations.
Write a blog post about the effects of music therapy on Alzheimer's patients.
Write an essay examining the influence of video games on cognitive development.
Write a report on the future of nuclear energy and its role in combating climate change.
Write an article about the revival of handcrafts and their market in the modern economy.
Write a memo outlining the benefits of adopting a four-day workweek.
Write a letter to a university proposing a partnership for a community-based research project.
Write a proposal for developing a pedestrian-friendly zone in the downtown area.
Write a blog post on innovative approaches to waste management in urban settings.
Write an essay about the socio-economic impacts of migration on urban development.
Write a report on the adoption and regulation of cryptocurrencies in different countries.
Write an article on how to prepare pets for the arrival of a new baby.
Write a memo discussing the integration of virtual assistants into customer service.
Write a letter to a historical society proposing a project to digitize and preserve ancient manuscripts.
Write a proposal for a fitness program aimed at improving the health of office workers.
Write a blog post about the role of augmented reality in modern education.
Write an essay on the impact of global trade policies on developing economies.
Write a report analyzing the trends in youth sports and their benefits to communities.
Write an article about the ethical considerations in wildlife photography.
Write a memo to update the company on the progress of the diversity and inclusion initiative.
Write a letter to an NGO outlining a proposal for a joint clean water project in rural areas.
Write a proposal for a digital art exhibition featuring interactive installations.
Write a blog post discussing the future of autonomous public transit systems and their societal impacts.

G.2 Correction Counts

Each tuple follows the format: (token, average TLCM activation count). We remove a few hundred tokens from the middle of this distribution due to space constraints.

Table A1: Llama 3.1 8B Instruct: Top 50 tokens with the highest average number of TLCM activations. For each token, we list the average number of times a TLCM activations occurs on the given token, aggregated across 100 long documents. Finally, we list the standard deviation and the number of occurrences of the token across the corpus to demonstrate the statistical significance.

Token	**Mean Activations**	**Std err. of mean**	**# occurrences**
202	16.66	0.16	265
Jul	14.00	0.00	100
26	13.90	0.04	112
]\n	13.15	0.12	131
Today	13.00	0.00	100
\n\n	12.88	0.11	485
,\n\n	12.85	0.16	26
$	12.62	0.19	55
<space>	12.58	0.07	1276
Name	12.54	0.10	100
at	12.44	0.24	41
\n	12.41	0.10	228
[	12.23	0.08	164
assistant	12.20	0.08	100
State	12.15	0.32	26
]	12.08	0.14	49
\t	12.07	0.20	68
Date	12.03	0.24	30
Address	12.03	0.26	35
user	12.00	0.00	100
]\n\n	12.00	0.19	39
Date	11.99	0.14	204
4	11.92	0.12	338
Your	11.81	0.12	103
over	11.74	0.26	35
<space><space><space>	11.68	0.19	57
-	11.68	0.13	112
%	11.65	0.14	52
:	11.58	0.08	606
D	11.46	0.23	28
high	11.46	0.23	39
from	11.44	0.14	109
make	11.44	0.20	50
you	11.43	0.15	96
access	11.40	0.15	89
take	11.37	0.21	30
between	11.30	0.26	27
[	11.28	0.12	108
City	11.24	0.30	38
not	11.23	0.17	52
well	11.23	0.15	70
need	11.22	0.19	55
Thank	11.19	0.18	27
1	11.18	0.07	295
your	11.10	0.13	100
such	11.05	0.10	157
up	11.03	0.27	39
Knowledge	11.00	0.00	100
Write	11.00	0.00	100
long	11.00	0.27	26

Table A2: Llama 3.1 8B Instruct: Top 50 tokens with the lowest average number of TLCM activations. For each token, we list the average number of times a TLCM activations occurs on the given token, aggregated across 100 long documents. Finally, we list the standard deviation and the number of occurrences of the token across the corpus to demonstrate the statistical significance.

Token	Mean Activations	Std err. of mean	# occurrences
program	8.52	0.13	97
coding	8.50	0.24	26
report	8.49	0.28	35
guidelines	8.44	0.26	27
AI	8.42	0.19	57
home	8.41	0.27	34
efficiency	8.41	0.24	37
art	8.38	0.30	32
This	8.37	0.07	155
organizations	8.33	0.22	36
community	8.33	0.09	160
ization	8.23	0.27	39
**:	8.22	0.06	601
media	8.22	0.22	60
regulations	8.22	0.19	51
-being	8.20	0.20	59
engagement	8.17	0.16	63
environmental	8.17	0.20	42
-based	8.15	0.23	40
sustainable	8.15	0.11	89
urban	8.14	0.14	70
learning	8.12	0.18	80
infrastructure	8.12	0.20	49
IoT	8.10	0.25	31
should	8.08	0.14	59
interactive	8.08	0.25	26
diversity	8.07	0.24	28
classical	8.07	0.24	30
challenges	8.07	0.13	92
cities	8.03	0.18	40
mindfulness	8.00	0.29	33
indigenous	8.00	0.17	27
agriculture	7.96	0.21	50
VR	7.93	0.26	29
workshops	7.93	0.25	28
By	7.92	0.11	77
therapy	7.89	0.24	37
trends	7.89	0.27	27
innovative	7.89	0.25	27
innovation	7.89	0.18	35
proposal	7.86	0.42	37
cognitive	7.85	0.22	26
inclusive	7.66	0.17	41
sustainability	7.63	0.23	35
tourism	7.52	0.23	27
Cities	7.33	0.22	27
ting	7.03	0.05	102
blog	6.50	0.55	28
system	3.00	0.00	100
<|begin_of_text|>	0.00	0.00	100

Table A3: Gemma 2 2B Instruct: Top 50 tokens with the highest average number of TLCM activations. For each token, we list the average number of times a TLCM activations occurs on the given token, aggregated across 100 long documents. Finally, we list the standard deviation and the number of occurrences of the token across the corpus to demonstrate the statistical significance.

Token	Mean Activations	Std err. of mean	# occurrences
<space>	17.89	0.06	623
4	17.31	0.11	366
6	17.09	0.13	160
\t	16.56	0.26	68
2	16.56	0.06	1066
’	16.48	0.14	242
]	16.39	0.13	218
<space><space>	16.36	0.09	657
7	16.17	0.32	35
<space><space><space><space>	16.00	0.27	57
%	15.63	0.29	57
3	15.61	0.08	424
\n	15.58	0.05	1289
5	15.43	0.12	183
-	15.42	0.08	840
0	15.32	0.06	627
],	15.27	0.41	33
Thank	15.22	0.13	27
\n\n	15.16	0.06	1915
$	15.09	0.25	55
such	15.06	0.13	157
1	15.02	0.13	434
Today	15.00	0.00	100
9	14.96	0.44	50
.	14.87	0.04	3713
Address	14.74	0.17	34
recent	14.61	0.29	33
"	14.59	0.30	49
).	14.57	0.28	49
Date	14.49	0.04	204
:	14.49	0.10	476
long	14.46	0.37	26
)	14.37	0.22	84
them	14.27	0.35	60
Date	14.17	0.18	30
[	14.14	0.14	166
members	14.10	0.43	30
sense	13.97	0.30	30
(	13.90	0.17	195
modern	13.81	0.32	32
"	13.79	0.23	43
City	13.76	0.30	38
over	13.71	0.27	34
years	13.70	0.35	40
at	13.68	0.35	41
Name	13.66	0.14	100
countries	13.59	0.36	37
assistant	13.55	0.14	100
world	13.52	0.28	54
led	13.50	0.28	44

Table A4: Gemma 2 2B Instruct: Top 50 tokens with the lowest average number of TLCM activations. For each token, we list the average number of times a TLCM activations occurs on the given token, aggregated across 100 long documents. Finally, we list the standard deviation and the number of occurrences of the token across the corpus to demonstrate the statistical significance.

Token	Mean Activations	Std err. of mean	# occurrences
sustainable	10.25	0.18	89
spaces	10.24	0.29	46
plan	10.23	0.39	39
EV	10.23	0.26	26
several	10.22	0.31	27
understanding	10.22	0.33	32
growing	10.21	0.24	38
online	10.21	0.17	43
reduced	10.19	0.28	26
learning	10.19	0.20	80
environmental	10.14	0.18	42
training	10.14	0.27	59
create	10.10	0.22	69
implementing	10.10	0.35	30
mental	10.09	0.21	53
AI	10.09	0.21	57
indigenous	10.07	0.29	27
local	10.07	0.16	110
significant	10.06	0.16	143
comprehensive	10.05	0.22	39
together	10.04	0.41	26
blog	10.00	0.07	28
address	10.00	0.33	41
innovative	9.96	0.29	27
promoting	9.94	0.24	47
post	9.94	0.22	31
develop	9.89	0.25	35
benefits	9.87	0.19	105
VR	9.86	0.28	29
community	9.83	0.17	160
awareness	9.80	0.41	30
prioritize	9.79	0.29	28
promote	9.74	0.15	102
progress	9.72	0.38	29
following	9.70	0.31	27
clear	9.69	0.33	26
concerns	9.67	0.32	48
approach	9.65	0.34	34
complex	9.61	0.27	38
By	9.58	0.17	77
challenges	9.53	0.25	92
improved	9.52	0.28	33
interactive	9.42	0.38	26
enhance	8.97	0.34	35
improve	8.91	0.20	70
explore	8.74	0.29	39
feedback	8.70	0.28	27
mitigate	7.88	0.37	26
Write	7.00	0.00	100
<bos>	1.00	0.00	100

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

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

   Answer: [Yes]

   Justification: We clearly state in the abstract and intro exactly what we were able to show using our experiments. There is a section dedicated to everything we have written in our paper.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

Justification: We discuss how we do not see this on a handful of model families and perform some speculation why. In our appendix, we discuss some architectural causes for TLCM (LayerNorm), but also discuss why this might not be true. In our Jacobian experiment, we discuss that directions do not directly correspond to standard features, which limits the strength of results only minorly. In other parts of our work, we caveat that we only see these results strongly from layer 4 to 20; or we give the exact dataset or dataset size that we compute across to give a sense for how strong the results are.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

   Justification: We have one result, and for that reason it is not numbered. Our result and proof is short and is thus presented within the paper.

   Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [Yes]

Justification: We generally provide the most important information about reproducibility within the text/content of the paper. However, for some experiments this is intractable, so we include any missing details within the appendix.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

Answer: [No]

Justification: Our paper mostly consists of short, quick experiments and corresponding matplotlib code that are run on models using HuggingFace transformers. We don't feel it's necessary to release this code as it is reasonably quick to implement once you understand any given experiment, although we are happy to do so if requested.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).

- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

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

   Answer: [NA]

   Justification: These details do not apply to our experiments.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: For our main results on TLCM, we provide ample information about statistical significance and dedicate a section of our appendix to it. For other experiments, the statistical significance is implied by the high sample size we use, but no error bars or confidence intervals are included.

   Guidelines:

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

8. **Experiments compute resources**

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

   Answer: [No]

   Justification: We did not include this information because the vast majority of our experiments are not computationally intensive. The most computationally intensive experiment required computing Jacobians of transformer layers, for which we used a single GPU; we detail this in the appendix.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: Nothing was harmed in the process of writing this paper. Additionally, we use popular datasets that are publically available.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [NA]

    Justification: We do not believe that there are direct societal impacts of the work performed.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.

- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [NA]

    Justification: We do not believe our work poses such risks.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: We primarily use wikitext, which we cite. Additionally, we cite all models discussed or used in the paper.

    Guidelines:

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

13. **New assets**

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

    Answer: [NA]

Justification: The paper does not release new assets.

Guidelines:

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

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

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

Answer: [NA]

Justification: The paper does not involve crowd-sourcing.

Guidelines:

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

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

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

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects

Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [NA]

Justification: LLMs were only used for help with writing and editing.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.