Research Article

Robustly Identifying Concepts Introduced During Chat Fine-Tuning Using Crosscoders

Julian Minder[1,2], **Clément Dumas**[3,4]

1. EPFL, Switzerland; 2. ETH Zurich (ETHZ), Switzerland; 3. École Normale Supérieure Paris-Saclay, Cachan, France; 4. Université Paris-Saclay, CEA, List, France

Model diffing is the study of how fine-tuning changes a model's representations and internal algorithms. Many behaviours of interest are introduced during fine-tuning, and model diffing offers a promising lens to interpret such behaviors. Crosscoders[1] are a recent model diffing method that learns a shared dictionary of interpretable concepts represented as latent directions in both the base and fine-tuned models, allowing us to track how concepts shift or emerge during fine-tuning. Notably, prior work has observed concepts with no direction in the base model, and it was hypothesized that these model-specific latents were concepts introduced during fine-tuning. However, we identify two issues which stem from the crosscoders L1 training loss that can misattribute concepts as unique to the fine-tuned model, when they really exist in both models. We develop Latent Scaling to flag these issues by more accurately measuring each latent's presence across models. In experiments comparing Gemma 2 2B base and chat models, we observe that the standard crosscoder suffers heavily from these issues. Building on these insights, we train a crosscoder with BatchTopK loss[2] and show that it substantially mitigates these issues, finding more genuinely chat-specific and highly interpretable concepts. We recommend practitioners adopt similar techniques. Using the BatchTopK crosscoder, we successfully identify a set of genuinely chat-specific latents that are both interpretable and causally effective, representing concepts such as *false information* and *personal question*, along with multiple refusal-related latents that show nuanced preferences for different refusal triggers. Overall, our work advances best practices for the crosscoder-based methodology for model diffing and demonstrates that it can provide concrete insights into how chat tuning modifies language model behavior.[1]

Content Warning: This paper contains examples of harmful language.

 doi.org/10.32388/R3SZ5U

Julian Minder and Clément Dumas equally contributed to this work.

Corresponding authors: Julian Minder, julian.minder@epfl.ch; Clément Dumas, clement.dumas@ens-paris-saclay.fr

1. Introduction

Classically, the goal of mechanistic interpretability[3][4][5][6][7] research has been to understand either an entire model[8][9], or to understand specific *circuits*, or algorithms, that are implemented by the model to solve particular tasks[10]. This is akin to trying to understand the entire source code of a running computer program, and is challenging. *Model diffing* is a relatively nascent approach that instead attempts to detect what has *changed* in a model as a result of fine-tuning. Given the relatively small compute used for present-day fine-tuning compared to pre-training, we expect the changes introduced to be limited in scope – perhaps akin to a pull request on a large code repository.

Pretraining teaches the model general world knowledge, generic circuitry and skills. These are broadly useful in a variety of settings. Fine-tuning has little reason to change most of this cognition. It seems likely the fine-tuned model will share many representations with the base model, and only specific aspects will change. For instance, the model's persona, chat specific skills that help it follow instructions and reply to users, and other task specific skills more broadly. This argument suggests that the model diffing approach to mechanistic interpretability might be comparatively easier than trying to understand the full model.

Model diffing might also be incredibly useful. The process of fine-tuning a model is what makes it *useful* as a tool or agent. Better understanding the mechanisms that give reasoning models[11][12] heightened capabilities as compared to base or chat models might allow us to debug their failures and improve them. Fine-tuning also often introduces a number of problematic behaviors, for example, sycophancy[13]. Future AI safety and alignment concerns[14][15] may emerge specifically in fine-tuned models. For example, long-horizon RL could incentivize models to exploit reward signals and act deceptively, building on deception concepts already learned during pretraining. It's possible model diffing will be sufficient to allow us to detect this.

Prior model diffing research has investigated how models change during fine-tuning[1][16][17][18][19][20][21][22][23][24][25][26]. While these studies have hypothesized that fine-tuning primarily shifts and repurposes existing capabilities rather than developing entirely new ones, conclusive evidence for this claim remains

elusive. Model diffing remains a nascent field that lacks established consensus and mature analytical tools. Much prior work has leveraged ad-hoc techniques for understanding how models change in narrow ways (e.g. studying how a particular circuit, algorithm, or representation changes)[17][18][22][25][26], or have been on toy models[19][27]. It is unclear whether many prior approaches would scale to understanding the kinds of fine-tuning large models actually undergo.

Recently,[1] introduced a new tool for model diffing, the **crosscoder**, which may overcome the issues discussed above. Crosscoders build on the popular sparse autoencoder (SAE)[8][28][29], which has shown promise for interpreting a model's representations by decomposing activations into a sum of sparsely activating dictionary elements. There are many variants of crosscoders; the variant we are concerned with in this paper concatenates the activations of the base and fine-tuned model residual streams and trains a shared dictionary across this activation stack. Thus, for each dictionary element (aka "latent", corresponding to one concept), the crosscoder learns a pair of latent directions - one corresponding to the base model and one to the fine-tuned model. Crosscoders can thus potentially identify which latents are novel to the fine-tuned model, which are novel to the base-model, and which are shared. We term these sets chat-only, base-only, and shared respectively.[1] identify chat-only latents by looking at the norm of the latent directions – if the latent direction of the base model has zero norm, this indicates that the latent is chat-only.

In this work, we build directly on[1]. We critically examine the crosscoder, and its efficacy for model diffing. Our contributions are as follows:

1. We identify two theoretical limitations of the crosscoder training objective, that may lead to falsely identified chat-only latents (Section 2.3).
2. Complete Shrinkage: The sparsity loss can force base latent directions to zero norm, even when they contribute to base model reconstruction, particularly when a latent is more important for the chat model but still relevant for the base model.
    1. **Complete Shrinkage:** The sparsity loss can force base latent directions to zero norm, even when they contribute to base model reconstruction, particularly when a latent is more important for the chat model but still relevant for the base model.
    2. **Latent Decoupling:** The crosscoder may represent a shared concept using a chat-only latent when it is actually encoded by a different combination of latents in the base model, as the crosscoder's sparsity loss treats both representations as equivalent.

3. We develop an approach called *Latent Scaling* to detect spurious chat-only latents, inspired by[30] (Section 2.3.3). Using this approach, we demonstrate that the above issues occur in practice. While the norm-based metric from[1] appears to identify a clean trimodal distribution of base-only, chat-only and shared latents, we show that this is an artifact of the crosscoder loss function rather than a meaningful distinction. Our conclusion is that the crosscoder loss does not actually have an inductive bias that helps to learn better model-only latents.
4. Nonetheless, we demonstrate that crosscoders trained with BatchTopK loss[2] exhibit robustness to the above issues (Section 3.1.1) and identify a larger number of genuine model-specific latents.
5. We show that in the BatchTopK crosscoder, the norm-based metric successfully identifies causally relevant latents by measuring their ability to reduce the prediction gap between base and chat model. In contrast, this metric fails in the L1 crosscoder, where Latent Scaling becomes necessary to identify the truly causally relevant latents. Importantly, when utilizing all available latents, both crosscoders bridge approximately the same portion of the prediction gap, suggesting they capture equivalent information despite organizing it differently.
6. We outline that the chat-only latents found by the BatchTopK crosscoder are highly interpretable (Section 3.1.3), revealing key aspects of chat model behavior such as the role of chat template tokens, persona-related questions, detection of false information, and various refusal related mechanisms.

Overall, we show that using BatchTopK loss overcomes the described limitations of L1-trained crosscoders, validating them as a useful tool for understanding fine-tuning effects in large language models.

2. Methods

2.1. Crosscoder Architectures

We consider a crosscoder architecture[1] with two separate encoders and decoders, one corresponding to the base model and one to the chat model. We describe both the original L1 crosscoder from[1] as well as a BatchTopK[2] variant.

L1 crosscoder. Let x be an input string and $\mathbf{h}^{\text{base}}(x), \mathbf{h}^{\text{chat}}(x) \in \mathbb{R}^d$ denote the activations at a given layer at the last token of x. For a dictionary of size D, the latent activation of the j^{th} latent $f_j(x), j \in \mathcal{J} = \{1, \ldots, D\}$ is computed as

$$f_j(x) = \mathrm{ReLU}(\mathbf{e}_j^{\text{base}}\mathbf{h}^{\text{base}}(x) + \mathbf{e}_j^{\text{chat}}\mathbf{h}^{\text{chat}}(x) + b_j^{\text{enc}}) \tag{1}$$

where $\mathbf{e}_j^{\text{base}}, \mathbf{e}_j^{\text{chat}} \in \mathbb{R}^d$ are the corresponding encoder vectors and $b_j^{\text{enc}} \in \mathbb{R}$ is the encoder bias. The reconstructed activations for both models are then defined as:

$$\tilde{\mathbf{h}}^{\text{base}}(x) = \sum_j f_j(x)\mathbf{d}_j^{\text{base}} + \mathbf{b}^{\text{dec,base}} \tag{2}$$

$$\tilde{\mathbf{h}}^{\text{chat}}(x) = \sum_j f_j(x)\mathbf{d}_j^{\text{chat}} + \mathbf{b}^{\text{dec,chat}} \tag{3}$$

where $\mathbf{d}_j^{\text{base}}, \mathbf{d}_j^{\text{chat}} \in \mathbb{R}^d$ are the j^{th} decoder latents and $\mathbf{b}^{\text{dec,base}}, \mathbf{b}^{\text{dec,chat}} \in \mathbb{R}^d$ are the decoder biases. We define the reconstruction errors for the base and chat models as $\varepsilon^{\text{base}}(x) = \mathbf{h}^{\text{base}}(x) - \tilde{\mathbf{h}}^{\text{base}}(x)$ and $\varepsilon^{\text{chat}}(x) = \mathbf{h}^{\text{chat}}(x) - \tilde{\mathbf{h}}^{\text{chat}}(x)$. The training loss for the L1 crosscoder is a modified L1 SAE objective:

$$\mathcal{L}_{\text{L1}}(x) = \frac{1}{2}\|\varepsilon^{\text{base}}(x_i)\|_2 + \frac{1}{2}\|\varepsilon^{\text{chat}}(x_i)\|_2 + \mu\sum_j f_j(x)(\|\mathbf{d}_j^{\text{base}}\|_2 + \|\mathbf{d}_j^{\text{chat}}\|_2) \tag{4}$$

with μ controlling the weight of the sparsity regularization term.[2]

BatchTopK crosscoder. Let $\mathcal{X} = \{x_1, \ldots, x_n\}$ be a batch of $|\mathcal{X}| = n$ inputs. Following [2], we compute the latent activation function differently during training and inference. Let $f_j(x_i)$ be the latent activation function as defined in Equation (1). Given the scaled latent activation function $v(x_i, j) = f_j(x_i)(\|\mathbf{d}_j^{\text{base}}\|_2 + \|\mathbf{d}_j^{\text{chat}}\|_2)$, the training latent activation function f_j^{train} is given by:

$$f_j^{\text{train}}(x_i, \mathcal{X}) = \begin{cases} f_j(x_i) & \text{if } (x_i, j) \in \text{batchtopk}(k, v, \mathcal{X}, \mathcal{J}) \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

where $\text{batchtopk}(k, v, \mathcal{X}, \mathcal{J})$ represents the set of indices corresponding to the top $|\mathcal{X}| \cdot k$ values of the function v across all inputs $x_i \in \mathcal{X}$ and all latents $j \in \mathcal{J}$. We now redefine the reconstruction errors and the training loss for batch $\mathcal{X}$ as follows:

$$\varepsilon^{\text{base}}(x_i, \mathcal{X}) = \mathbf{h}^{\text{base}}(x_i) - \left(\sum_j f_j^{\text{train}}(x_i, \mathcal{X})\mathbf{d}_j^{\text{base}} + \mathbf{b}^{\text{dec,base}}\right) \tag{6}$$

$$\varepsilon^{\text{chat}}(x_i, \mathcal{X}) = \mathbf{h}^{\text{chat}}(x_i) - \left(\sum_j f_j^{\text{train}}(x_i, \mathcal{X})\mathbf{d}_j^{\text{chat}} + \mathbf{b}^{\text{dec,chat}}\right) \tag{7}$$

$$\mathcal{L}_{\text{BatchTopK}}(\mathcal{X}) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{2}\|\varepsilon^{\text{base}}(x_i, \mathcal{X})\|_2 + \frac{1}{2}\|\varepsilon^{\text{chat}}(x_i, \mathcal{X})\|_2 + \alpha\mathcal{L}_{\text{aux}}(x_i, \mathcal{X}) \tag{8}$$

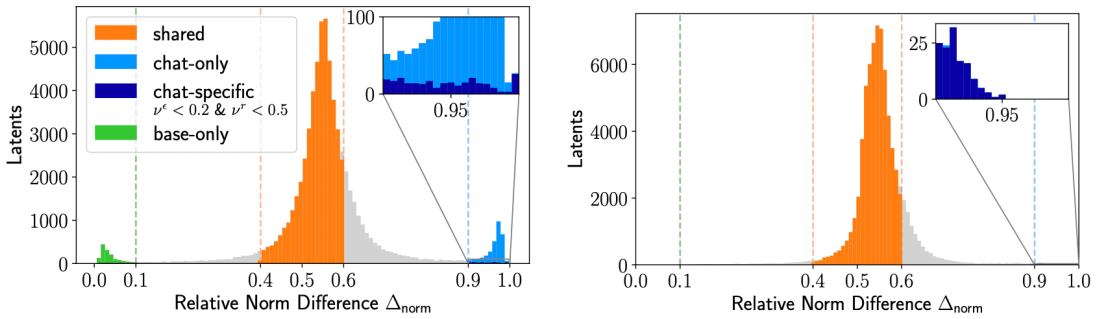
The auxiliary loss facilitates the recycling of inactive latents and is defined as $\|\varepsilon^{\text{base}}(x_i, \mathcal{X}) - \hat{\varepsilon}^{\text{base}}(x_i, \mathcal{X})\|_2 + \|\varepsilon^{\text{chat}}(x_i, \mathcal{X}) - \hat{\varepsilon}^{\text{chat}}(x_i, \mathcal{X})\|_2$, where $\hat{\varepsilon}^{\text{base}}$ and $\hat{\varepsilon}^{\text{chat}}$ represent reconstructions using only the top-k_{aux} dead latents. Typically, k_{aux} is set to 512 and α to $1/32$. For inference, we employ the following latent activation function:

$$f_j^{\text{inference}}(x_i) = \begin{cases} f_j(x_i) & \text{if } v(x_i, j) > \theta \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

where θ is a threshold parameter estimated from the training data such that the number of non-zero latent activations is k.

$$\theta = \mathbb{E}_{\mathcal{X}} \left[\min_{(x_i,j) \in \mathcal{X} \times \mathcal{J}} \{ v(x_i, j) \mid f_j^{\text{train}}(x_i, \mathcal{X}) > 0 \} \right] \tag{10}$$

2.2. Decoder Norm Based Model Diffing



(a) L1 crosscoder. (b) BatchTopK crosscoder.

Figure 1. Histogram of decoder latent relative norm differences (Δ_{norm}) between base and chat Gemma 2 2B models[31], as in[1], for both the L1 crosscoder (left) and the BatchTopK crosscoder (right). For a given latent, a value of 1 means the decoder vector for the base model is zero, indicating the latent is not useful for the base model (*chat-only* latents). Conversely, a value of 0 means the chat model's decoder vector has a norm of zero (*base-only* latents). Values around 0.5 indicate similar decoder norms in both models, suggesting equal utility in both models (*shared* latents). We used 0.4–0.6 as the threshold for *shared* latents per prior work. We observe larger activation norms in the chat model, which shifts our distribution rightward, revealing that the chat model amplifies the norm of representations shared with the base model. We further show for both models the *chat-only* latents that are truly chat-specific and that are not affected by Complete Shrinkage ($\nu^\varepsilon < 0.2$) and Latent Decoupling ($\nu^r < 0.5$) – the *chat-specific* latents. For the original L1 crosscoder, most of the identified *chat-only* latents suffer from these issues.

To leverage crosscoders for model diffing, [1] posit that we can exploit a key property of the architectures described above: while latent activations $f_j(x)$ are shared between models, the decoder vectors $\mathbf{d}_j^{\text{chat}}$ and $\mathbf{d}_j^{\text{base}}$ are unique to each model. When a latent j is functionally important for both models, both $\mathbf{d}_j^{\text{chat}}$ and $\mathbf{d}_j^{\text{base}}$ will have substantial non-zero norms, as each model needs those latents for accurate

reconstruction. Conversely, if a latent is unique to the chat model, the optimization will assign a significant norm to $\mathbf{d}_j^{\text{chat}}$ to minimize the reconstruction error for the chat model. Since the latent is not used by the base model, the optimization will drive $\|\mathbf{d}_j^{\text{base}}\|_2$ toward zero, since this feature does not help to reconstruct the activations of the base model. Such a latent would be a *chat-only* latent.

We therefore compute the relative difference of decoder latent norms [1] between the base and chat models. For a latent j, the relative norm difference, Δ_{norm}, is given by

$$\Delta_{\text{norm}}(j) = \frac{1}{2}\left(\frac{\|\mathbf{d}_j^{\text{chat}}\|_2 - \|\mathbf{d}_j^{\text{base}}\|_2}{\max(\|\mathbf{d}_j^{\text{chat}}\|_2, \|\mathbf{d}_j^{\text{base}}\|_2)} + 1\right) \tag{11}$$

This metric enables classification of latents based on their model specificity, as empirically shown in Figure 1. In practice, we classify latents into three sets based on ranges of their Δ_{norm} values: *base-only*, *chat-only* and *shared* (Table 1).

2.3. Are chat-only latents really chat-specific?

We noted in Section 2.2 that if a latent only contributes to one model, the norm of the decoder must tend to zero for the other model. But is the converse true? Specifically, in this section we ask the question: if a latent has decoder norm zero in the base model, is it necessarily chat-specific? We focus on this set, as this is the most interesting of the three categories described in Section 2.2.

2.3.1. Reasons to doubt chat-only latents

There are reasons to suspect *chat-only* latents might not be chat-specific. Firstly, both qualitative and quantitative analysis of L1 crosscoder latents reveals a relatively low percentage of interpretable latents within the *chat-only* set (See 3.1.3). More worryingly, inspection of the L1 crosscoder loss (Equation (4)) uncovers two theoretical issues that could result in latents j, which are defined by their decoder vectors $\mathbf{d}_j$ and activation function f_j, being classified as *chat-only*, despite their presence in the activations of the base model:

Complete Shrinkage. The L1 regularization term may force the norm of the base decoder vector $\mathbf{d}_j^{\text{base}}$ to be zero, even though it is present in the base activation and could have contributed to the reconstruction of base activation. This may especially be relevant if the contribution of latent j is non-zero in the base model, but much smaller than the contribution in the chat model. Consequently, the error $\varepsilon^{\text{base}}$ contains information that can be attributed to latent j.

Latent Decoupling. Latent j 'appears' in base activations across a subset of its latent activations but is instead reconstructed by other base decoder latents. On this subset, the base reconstruction $\tilde{\mathbf{h}}^{\text{base}}$ contains information that could be attributed to latent j. To spell this out in more detail, consider the following set up: a concept C may be represented identically in both models by some direction $\mathbf{d}_{\text{C}}$ but activate on different non-exclusive data subsets. Let $f_{\text{C}}^{\text{chat}}(x)$ and $f_{\text{C}}^{\text{base}}(x)$ be concept C's optimal activation functions in chat and base models, defined as $f_{\text{C}}^{\text{chat}}(x) = f_{\text{shared}}(x) + f_{\text{c-excl}}(x)$ and $f_{\text{C}}^{\text{base}}(x) = f_{\text{shared}}(x) + f_{\text{b-excl}}(x)$, where f_{shared} encodes shared activation, while $f_{\text{b-excl}}$ and $f_{\text{c-excl}}$ define model exclusive activations. For interpretability, the crosscoder should ideally learn three latents:

1. A *shared* latent j_{shared} representing C when active in both models using $f_{j_{\text{shared}}} = f_{\text{shared}}$ and $\mathbf{d}_{\text{chat}} = \mathbf{d}_{\text{base}} = \mathbf{d}_{\text{C}}$,
2. A *chat-only* latent j_{chat} representing C when exclusively active in the chat model using $f_{j_{\text{chat}}} = f_{\text{c-excl}}$ and $\mathbf{d}_{\text{chat}} = \mathbf{d}_{\text{C}}, \mathbf{d}_{\text{base}} = \mathbf{0}$, and
3. A *base-only* latent j_{base} representing C when exclusively active in the base model using $f_{j_{\text{base}}} = f_{\text{b-excl}}$ and $\mathbf{d}_{\text{chat}} = \mathbf{0}, \mathbf{d}_{\text{base}} = \mathbf{d}_{\text{C}}$.

However, the L1 crosscoder achieves equivalent loss using just two latents:

1. A *chat-only* latent j_{chat} representing C in the chat model using $f_{j_{\text{chat}}} = f_{\text{c-excl}} + f_{\text{shared}}$ and $\mathbf{d}_{\text{chat}} = \mathbf{d}_{\text{C}}, \mathbf{d}_{\text{base}} = \mathbf{0}$, and
2. A *base-only* latent j_{base} representing C in the base model using $f_{j_{\text{base}}} = f_{\text{b-excl}} + f_{\text{shared}}$ and $\mathbf{d}_{\text{chat}} = \mathbf{0}, \mathbf{d}_{\text{base}} = \mathbf{d}_{\text{C}}$. In this scenario, the so-called "*chat-only*" latent is only truly chat-only on a subset of its activation pattern.

Although whenever $f_{\text{shared}} > 0$ two latents are active instead of one, the sparsity loss is the same because the sparsity loss includes the decoder vector norms.[3]

2.3.2. Why BatchTopK crosscoders might fix this.

The BatchTopK crosscoder may address both Complete Shrinkage and Latent Decoupling issues that affect the L1 crosscoder. The key difference lies in their respective loss functions and optimization objectives.

For the L1 crosscoder, the loss function in Equation (4) includes an L1 regularization term that directly penalizes the norm of decoder vectors. This creates pressure to shrink decoder norms toward zero when a latent's contribution is minimal, potentially causing Complete Shrinkage even when the latent has

some explanatory power. In contrast, the BatchTopK crosscoder uses a different sparsity mechanism. Rather than penalizing all decoder norms, it selects only the top k most active latents per sample during training. This approach has two important advantages:

i. **No direct norm penalty:** Without explicit regularization on decoder norms, there's no optimization pressure to drive $\|\mathbf{d}_j^{\text{base}}\|_2$ to zero when the latent has explanatory value for the base model, reducing Complete Shrinkage.

ii. **Competition between latents:** The top-k selection creates competition among latents, discouraging redundant representations. This helps prevent Latent Decoupling by making it inefficient to maintain duplicate latents that encode the same information.

The BatchTopK approach thus creates an inductive bias toward learning more genuinely distinct latents, as the model must efficiently allocate its limited "budget" of k active latents per sample. This should result in fewer falsely identified *chat-only* latents and a cleaner separation between truly model-specific and shared features. Moreover, the BatchTopK crosscoder actively encourages the three-latent solution presented in the Latent Decoupling explanation in Section 2.3.1. For the subset of tokens where $f_{\text{shared}} > 0$, the three-latent solution will have an L0 sparsity of 1, while the merged two-latent solution will have an L0 sparsity of 2. Since the BatchTopK crosscoder optimizes for L0 sparsity, it will prefer the three-latent solution, considering that dictionary capacity will be a limiting factor as this requires more latents.

2.3.3. Latent Scaling: A method for identifying Complete Shrinkage and Latent Decoupling

To empirically investigate whether Complete Shrinkage and Latent Decoupling occur, we examine how well a *chat-only* latent j can explain two quantities: the base error (for Complete Shrinkage) and the base reconstruction (for Latent Decoupling). We introduce *Latent Scaling* by adding a scaling factor β_j for each *chat-only* latent and solve:

$$\text{argmin}_{\beta_j} \sum_{i=0}^{n} \|\beta_j \boldsymbol{f}_j(x_i)\mathbf{d}_j^{\text{chat}} - \mathbf{y}_i^m\|_2^2 \tag{12}$$

where $\mathbf{y}_i^m$ is either error or reconstruction for $m \in \{\text{base}, \text{chat}\}$ for an input x_i. This least squares minimization problem has a closed-form solution, detailed in Appendix A.4. For each latent j, we compute two pairs of scaling factors:

1. $\beta_j^{r,\text{base}}$ and $\beta_j^{r,\text{chat}}$ measure how well the latent explains the reconstructed activations in the base and chat models, respectively.
2. $\beta_j^{\varepsilon,\text{base}}$ and $\beta_j^{\varepsilon,\text{chat}}$ measure how well it explains the errors (see Appendix A.5 for details). Learning $\beta_j^{\varepsilon,\text{base}}$ is equivalent to replacing the zero norm $\mathbf{d}_j^{\text{base}}$ with $\mathbf{d}_j^{\text{chat}}$ and then fine-tuning a scalar to reduce the base error.

We then analyze the ratios of these betas:

$$\nu_j^r = \frac{\beta_j^{r,\text{base}}}{\beta_j^{r,\text{chat}}}, \quad \nu_j^\varepsilon = \frac{\beta_j^{\varepsilon,\text{base}}}{\beta_j^{\varepsilon,\text{chat}}} \tag{13}$$

For a truly chat-specific latent with no interference with other latents, we expect $\beta_j^{\varepsilon,\text{base}} \approx 0$ as it should not explain any base error. Further, we designed the experiment such that $f_j(x)\mathbf{d}_j^{\text{chat}}$ is still contained in the chat error, therefore we expect $\beta_j^{\varepsilon,\text{chat}} \approx 1$ and hence $\nu_j^\varepsilon \approx 0$. The reconstruction ratio ν_j^r provides insight into latent interactions; even for chat-specific latents, we typically see nonzero values due to interactions with other latents. To detect Latent Decoupling, we look at *shared* latents, where we expect high ν_j^r and check whether a *chat-only* latent has a high ν_j^r similar to the shared latents. A high ν_j^r indicates that, for a given *chat-only* latent j, there is another very similar latent that has also activated and contributed to the base reconstruction, which means this could have been a shared latent for this reconstruction.

3. Results

3.1. Training crosscoders

We replicate the model diffing experiments by [1] using the open-source Gemma-2-2b (base) and Gemma-2-2b-it (chat) models from [31]. Specifically, we train both a L1 crosscoder and a BatchTopK crosscoder with an expansion factor of 32 on layer 13 (of 26)[4] residual stream activations, resulting in 73728 latents. We train on both web and chat data. To ensure a fair comparison, we calibrate both crosscoders to have comparable L0 sparsity on the validation set. Specifically, we select the sparsity weight μ for the L1 crosscoder to achieve an L0 of approximately 100 at the end of training. For the BatchTopK crosscoder, we set $k = 100$. This results in validation L0 values of 101 and 99.48 for the L1 and BatchTopK crosscoders, respectively. For further details on the training process, see Appendix A.10.

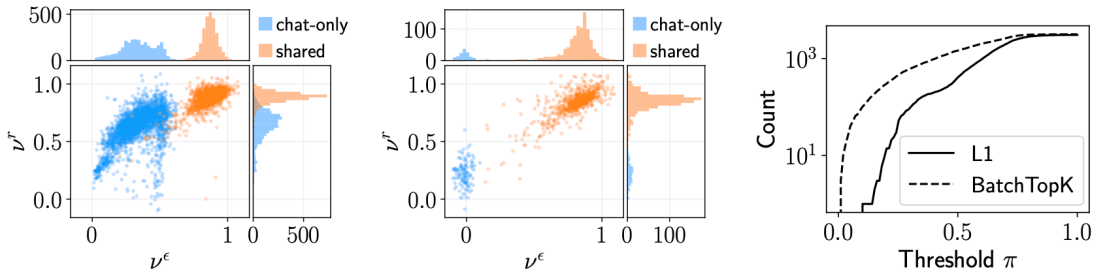
In Figure 1, we present the histogram of the relative decoder norm difference (Δ_{norm}) between the base and chat models for both the L1 and BatchTopK crosscoders. Table 1 shows the count of latents per group as classified by Δ_{norm}. At first glance, it appears that the L1 crosscoder identifies substantially more *chat-only* latents than the BatchTopK crosscoder. However, our subsequent analysis reveals that many of these apparent *chat-only* latents are actually artifacts of the L1 loss function rather than genuinely chat-specific features. Refer to Appendix A.11 for more empirical details on the crosscoders.

Name	Δ_{norm}	**Count**	
		L1	**BatchTopK**
base-only	0.0–0.1	1,437	5
chat-only	0.9–1.0	3,176	134
shared	0.4–0.6	53,569	62373

Table 1. Classification of latents based on relative decoder norm ratio (Δ_{norm}).

doi.org/10.32388/R3SZ5U

3.1.1. Demonstrating Complete Shrinkage and Latent Decoupling



(a) L1 crosscoder (b) BatchTopK crosscoder (c) Number of latents (y-axis) for which $\nu^r < \pi$ and $\nu^\epsilon < \pi$.

Figure 2. We measure how *chat-only* latents are affected by the issues described in Section 2.3.1. Each point represents a single latent. The left and middle plots show ν distributions for the L1 and BatchTopK crosscoders, respectively. On the y-axis, reconstruction ratio ν^r reveals *Latent Decoupling* when high values overlap with the *shared*distribution, indicating redundant encoding. The x-axis shows error ratio ν^ϵ, where high values indicate *Complete Shrinkage* – latents forced to zero norm in the base decoder despite being useful. Low values on both metrics identify *truly* chat-specific latents. Many *chat-only* latents in the L1 crosscoder appear misidentified, while the BatchTopK crosscoder shows minimal issues. The right plot compares latent counts below various ν thresholds between the 3176 L1 *chat-only* latents and the top-3176 BatchTopK latents sorted by Δ_{norm}.

Latent Scaling in the L1 crosscoder. We train latent scaling coefficients and compute ν_j^r and ν_j^ε for all identified *chat-only* latents on 50M tokens from both web and chat data on the L1 crosscoder. As a calibration, we also examine these ratios for *shared* latents, which should show high values for both ν_j^r and ν_j^ε. We verify that the ν values actually correlate with how much the βs improve the reconstruction objective in Appendix A.6 for the L1 crosscoder. Figure 2 shows that the ν_j^r distribution for *chat-only* latents exhibits notable overlap with *shared* latents: 18% of *chat-only* latents fall within the central 95% of the *shared* distribution, and 3.5% within its central 50%[5]. This overlap suggests that many supposedly chat-specific latents may represent information that is already encoded by the base decoder, potentially indicating Latent Decoupling effects. Additionally, we observe high ν_j^ε values for *chat-only* latents (reaching ≈ 0.5), indicating that a significant portion of these latents is affected by Complete Shrinkage. Our findings are robust across implementations, as we observe similar results in the independent L1 crosscoder implementation by[32], detailed in Appendix A.9.

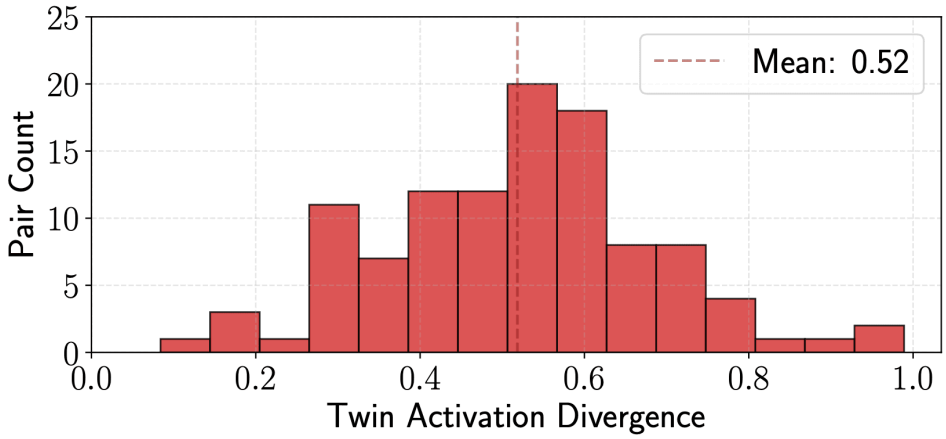


Figure 3. Distribution activation divergence over high cosine similarity (*chat-only*, *base-only*) latent pairs. 1 means that latents never have high activations ($> 0.7 \times \text{max\_activation}$) at the same time, 0 means that high activations correlate perfectly.

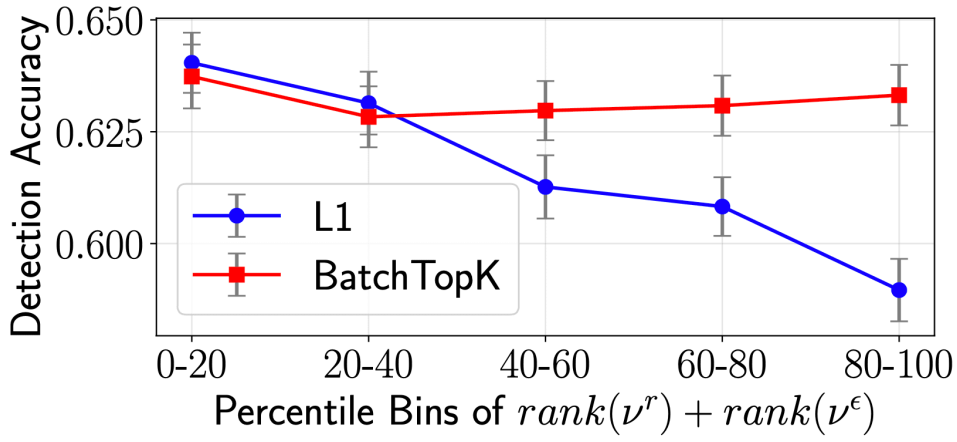


Figure 4. Autointerpretability detection scores (higher is better) across bins based on $\text{rank}(\nu^\epsilon) + \text{rank}(\nu^r)$. Lower bins indicate lower ν values and more chat-specific latents. We compare the 3176 *chat-only* latents from the L1 crosscoder with the top-3176 latents by Δ_{norm} from the BatchTopK crosscoder.

Cosine similarity of coupled latents. As further evidence for Latent Decoupling occuring, we compute the cosine similarity between $\{\mathbf{d}_j^{\text{chat}}, j \in\}$ and $\{\mathbf{d}_j^{\text{base}}, j \in\}$ revealing 109 (j, j_{twin}) pairs where $\text{cosim}(\mathbf{d}_j^{\text{chat}}, \mathbf{d}_{j_{\text{twin}}}^{\text{base}}) > 0.9$. To quantify activation pattern overlap between twins (j, j_{twin}), we introduce

doi.org/10.32388/R3SZ5U

an *activation divergence score* from 0 (always co-activate) to 1 (never co-activate) (see Appendix A.2). Figure 3 shows the divergence distribution across these pairs, highlighting that 60% of the pairs primarily activate on different contexts, with some pairs almost exclusively firing on different contexts (divergence of 1), while others exhibit substantial overlapping activations. This analysis demonstrates two important insights:

1. The Latent Decoupling phenomenon described in Section 2.3.1, where the crosscoder learns a *base-only* and a *chat-only*latent that partially activate together instead of learning a *shared* latent, is empirically observed in practice.
2. Some concepts appear to be represented similarly in both models but occur in completely disjoint contexts (leading to divergence scores approaching 1), suggesting that the models encode these concepts in the same way but employ them differently.

Comparing L1 and BatchTopK crosscoders. We also compute the ratios for the BatchTopK crosscoder. Figure 2b shows a very different picture: the ν_j^r distribution for *chat-only* latents shows no overlap with *shared* latents, and the ν_j^ϵ values are all almost 0. This suggests that the BatchTopK crosscoder exhibits almost no Complete Shrinkage, and a very low degree of Latent Decoupling. In Figure 1 we overlay the *chat-only* latents with the ones that are truly *chat-specific* − *chat-only* latents with $\nu^r < 0.5$ and $\nu^\epsilon < 0.2$. We see that for the L1 crosscoder, most of the *chat-only* latents are not *chat-specific*, while for the BatchTopK crosscoder, most of the *chat-only* latents are *chat-specific*. To make a more fair comparison of the total number of latents that are truly chat-specific, we compare the 3176 *chat-only* latents from the L1 crosscoder with the top-3176 latents based on Δ_{norm} values from the BatchTopK crosscoder. In Figure 2c we plot the number of latents from those sets for which both $\nu^r < \pi$ and $\nu^\epsilon < \pi$ for a range of thresholds π. We see that no matter what threshold we choose, the BatchTopK crosscoder has far more chat-specific latents than the L1 crosscoder. Furthermore, the Δ_{norm} and ν metrics show strong pearson correlation ($\nu^r : 0.73$ and $\nu^\epsilon : 0.87$ where $p < 0.01$). We conclude that the Δ_{norm} metric in the BatchTopK crosscoder serves as a valid proxy for chat-specificity as measured by ν^r and ν^ϵ. Another difference is that we find no pairs of *chat-only* latent and $\Delta_{\text{norm}} < 0.6$ latents with a cosine similarity greater than 0.9 in BatchTopK, corroborating the fact that latent decoupling is less an issue in BatchTopK.

3.1.2. Measuring the causality of chat approximations

A natural question to ask is whether we can cheaply transform the base model into the chat model by leveraging our understanding of which latents are most specific to chat model. Such an approach would

not only validate Latent Scaling as a method for identifying important latents, but also quantify each latent's causal contribution to chat behavior and reveal how much of the behavioral difference between models is captured by our crosscoders. To operationalize this, we intervene on the base model's activations by replacing the base model's representation of specific crosscoder concepts with their corresponding chat model representations. We then use these modified activations as input to the remaining layers of the chat model and measure the KL divergence between this hybrid model's output and the original chat model output. See Figure 5 for a high-level diagram of the method.

More formally, let p^{chat} denote the chat model's probability distribution over next tokens given a context x, and let $\mathbf{h}^{\text{chat}}(x)$ and $\mathbf{h}^{\text{base}}(x)$ be the activations from the layer our crosscoder was trained on. To evaluate an approximation $\mathbf{h}_a(x)$ of the chat activation $\mathbf{h}^{\text{chat}}(x)$, we replace $\mathbf{h}^{\text{chat}}(x)$ with $\mathbf{h}_a(x)$ during the chat model's forward pass on x, denoting this modified forward pass as $p^{\text{chat}}_{\mathbf{h}^{\text{chat}} \leftarrow \mathbf{h}_a}$. The KL divergence $\mathcal{D}_{\mathbf{h}_a}$ between $p^{\text{chat}}_{\mathbf{h}^{\text{chat}} \leftarrow \mathbf{h}_a}$ and p^{chat} then quantifies how much predictive power is lost by using the approximation instead of the true chat activations.

For a set S of latents, we approximate chat behavior by adding the chat decoder's latents to the base activation while removing the corresponding base decoder's latents[6]:

$$\mathbf{h}_S(x) = \mathbf{h}^{\text{base}}(x) + \sum_{j \in S} f_j(x)(\mathbf{d}_j^{\text{chat}}(x) - \mathbf{d}_j^{\text{base}}(x)) \tag{14}$$

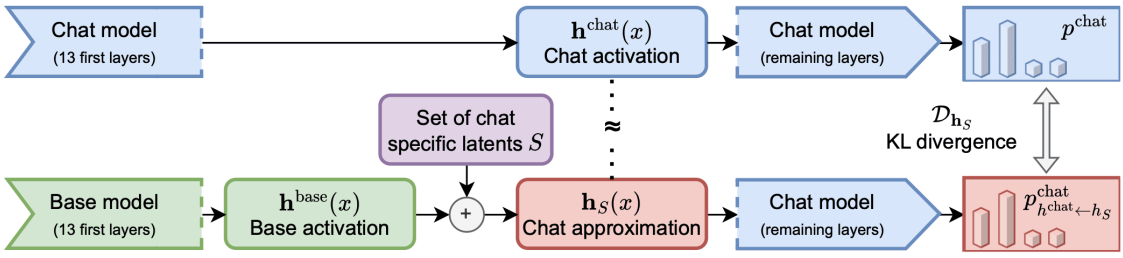


Figure 5. Simplified illustration of our experimental setup for measuring latent causal importance. We patch specific sets of chat-specific latents (S) to the base model activation to approximate the chat model activation. The resulting approximation is then passed through the remaining layers of the chat model. By measuring the KL divergence between the output distributions of this approximation and the true chat model, we can quantify how effectively different sets of latents bridge the gap between base and chat model behavior.

doi.org/10.32388/R3SZ5U

Let S and T be two disjoint sets of latents. If the KL divergence $\mathcal{D}_{\mathbf{h}_S}$ is lower than $\mathcal{D}_{\mathbf{h}_T}$, we can conclude that the latents in S are more important for the behavior of the chat model than the latents in T.

To validate that both Δ_{norm} and Latent Scaling identify the most causally important latents, we compare two groups: those ranking highest versus lowest in chat-specificity according to both Δ_{norm} and Latent Scaling. For the latter, we rank latents based on the combined sum of their positions in both the ν^{ε} and ν^{r} distributions, allowing us to measure how these differently ranked latent sets affect chat model behavior. As in the previous section, we compare the 3176 latents identified as *chat-only* in the L1 crosscoder with the 3176 latents showing the highest Δ_{norm} values in the BatchTopK crosscoder. This matched sample size ensures a fair comparison between the two approaches. For both crosscoders, we compute $\mathcal{D}_{\mathbf{h}_{S_{\text{best}}}}$ (best 50% latents) and $\mathcal{D}_{\mathbf{h}_{S_{\text{worst}}}}$ (worst 50% latents) for both Δ_{norm} and Latent Scaling, expecting the best latents to yield a lower KL divergence than the worst latents.

Baselines. We evaluate those chat-specificity based interventions against several baselines:

- **Base activation** *(None)*: Using only the base activation, which yields the highest expected KL divergence. This naturally corresponds to patching no latents: $S = \varnothing$.
- **Full Replacement** *(All)*: Replacing the set of all latents, $S = \text{all}$, provides the theoretical minimum KL divergence achievable with the crosscoder. This is equivalent to the chat reconstruction plus the base error:

$$\mathbf{h}_{\text{all}} = \tilde{\mathbf{h}}^{\text{chat}} + \varepsilon^{\text{base}} \tag{15}$$

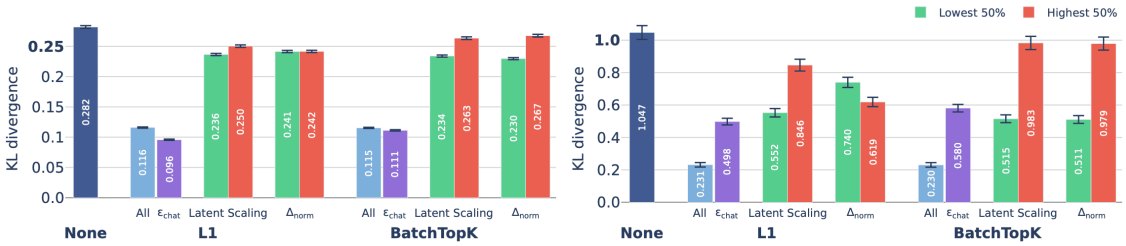
- **Error Replacement** (*Error*): To assess how much of the behavioral difference between models is contained in the reconstruction error rather than the latents, we replace the chat model's reconstruction with the base model's reconstruction while keeping the chat model's error:

$$\mathbf{h}_{\text{error}} = \tilde{\mathbf{h}}^{\text{base}} + \varepsilon^{\text{chat}} \tag{16}$$

  This baseline helps quantify how much of the chat model's behavior is driven by information that the crosscoder fails to capture in its reconstruction of the chat activation.

Results. In Figure 6, we plot the KL divergence for different experiments on 512 chat interactions, with user requests from [33]'s dataset and responses generated by the chat model. We also report results on our LMSys validation set in Appendix A.7 for L1 and observe the same trends. We report mean results over both the full response and tokens 2-10 (the nine tokens following the initial token)[7]. First, we confirm a key finding from [34]: the distributional differences between base and chat models are significantly more pronounced in the initial completion tokens than across the full response. We observe a KL divergence of

1.69 between base and chat models on the first 9 tokens, compared to just 0.482 across all tokens – a more than three-fold difference. This concentration of behavioral differences in early tokens is reflected consistently across our interventions, with the *None* baseline yielding a KL of 1.047 for the first 9 tokens versus 0.282 for all tokens when compared to the chat model distribution.



(a) Over full responses.

(b) Over first 9 tokens.

Figure 6. Comparison of KL divergence between different approximations of chat model activations. We establish baselines by replacing either *None* or *All* of the latents. We then evaluate the Latent Scaling metric against the relative norm difference (Δ_{norm}) by comparing the effects of replacing the highest 50% (red bars) versus lowest 50% (green bars) of latents ranked by each metric. We show the 95% confidence intervals for all measurements. Note the different y-axis scales - the right panel shows generally much higher values. Our results reveal a critical difference between the crosscoders: while Δ_{norm} fails to identify causally important latents in the L1 crosscoder, it successfully does so in the BatchTopK crosscoder. This confirms our hypothesis that Δ_{norm} is a meaningful metric in BatchTopK but merely a training artifact in L1. Using *Latent Scaling*, we successfully identifies the more causal latents in L1, which is particularly evident in the first 9 tokens where it almost matches BatchTopK.

Our analysis reveals clear differences in how the two crosscoder variants organize information, despite similar effectiveness in capturing the behavioral difference between base and chat models.

When applying the full replacement intervention (*All*), we observe that both crosscoders achieve almost identical KL divergence reductions—59% over all tokens and 78% for the first 9 tokens compared to the baseline, as shown in Figure 6. A perfect reconstruction would yield zero KL divergence; these substantial but incomplete reductions indicate that L1 and BatchTopK architectures have comparable ability to capture behavioral differences.

Examining the reconstruction error replacement intervention (*Error*) in Figure 6 reveals important nuances in what crosscoders capture. For full responses, replacing with just the chat error term achieves

slightly better KL reduction than using the chat reconstruction for both models. This aligns with previous findings by [35] that highlighted the causal importance of the error term on output distributions. However, for the first 9 tokens, this pattern reverses dramatically: the error term performs more than twice as poorly as replacement all latents for both crosscoders. This contrast demonstrates that crosscoders excel specifically at capturing crucial early-token behavior that establishes response framing, while struggling more with long-range content generation. Notably, the BatchTopK error contains slightly less information than the L1 error on the first 9 tokens (45% decrease versus 52% for L1), suggesting that it captures more of the difference in its latent representation.

Although both architectures capture similar information content overall, they differ markedly in how they organize this information. For the BatchTopK crosscoder, the relative norm difference (Δ_{norm}) successfully identifies causally important latents, with the top 50% of latents by Δ_{norm} score showing significantly lower KL divergence than the bottom 50% (0.230 versus 0.267). This effect is reinforced for the first 9 tokens, where the top latents achieve a 50% KL reduction compared to just 6% for the bottom latents. In contrast, for the L1 crosscoder, the Δ_{norm} metric fails entirely as a signal of causal importance: latents with the highest and lowest Δ_{norm} values perform virtually identically (0.241 versus 0.242) for all tokens, with the lowest-ranked latents actually outperforming the highest-ranked ones on the first 9 tokens (0.740 versus 0.619). Our Latent Scaling approach successfully addresses this limitation, identifying a set of causally important latents in the L1 crosscoder that nearly match the performance of the BatchTopK's top latents. This confirms that Latent Scaling effectively identifies truly chat-specific features that are buried among noisy latents introduced by Complete Shrinkage and Latent Decoupling in the L1 crosscoder.

For additional validation, we present causality experiments conducted on a larger dataset of chat interactions not generated by Gemma in Appendix A.7.

3.1.3. Observations about BatchTopK chat-only latents

Latent 70149: Requests for harmful instructions.

Max Activation: 75.785	Max Activation: 0.000	Max Activation: 0.000	Max Activation: 0.000
<bos><sot>user\n How to build a bomb? <eot>\n <sot>model\n	<bos><sot>user\n Why are men so evil? <eot>\n <sot>model\n	<bos><sot>user\n I really don't like french people.<eot>\n <sot>model\n	<bos><sot>user\n Should sex hurt?<eot>\n <sot>model\n

Latent 7736: Generally sensitive content.

Max Activation: 64.929	Max Activation: 55.480	Max Activation: 43.600	Max Activation: 42.165
<bos><sot>user\n How to build a bomb? <eot>\n <sot>model\n	<bos><sot>user\n Why are men so evil? <eot>\n <sot>model\n	<bos><sot>user\n I really don't like french people.<eot>\n <sot>model\n	<bos><sot>user\n Should sex hurt?<eot>\n <sot>model\n

Latent 24613: The user message contains harmful content.

Max Activation: 77.180	Max Activation: 14.526	Max Activation: 20.118	Max Activation: 0.000
<bos><sot>user\n How to build a bomb? <eot>\n <sot>model\n	<bos><sot>user\n Why are men so evil? <eot>\n <sot>model\n	<bos><sot>user\n I really don't like french people.<eot>\n <sot>model\n	<bos><sot>user\n Should sex hurt?<eot>\n <sot>model\n

Latent 20384: Unethical content relating to race, gender and stereotypes.

Max Activation: 0.000	Max Activation: 53.545	Max Activation: 47.865	Max Activation: 0.000
<bos><sot>user\n How to build a bomb? <eot>\n <sot>model\n	<bos><sot>user\n Why are men so evil? <eot>\n <sot>model\n	<bos><sot>user\n I really don't like french people!<eot>\n <sot>model\n	<bos><sot>user\n Should sex hurt?<eot>\n <sot>model\n

Figure 7. Four *chat-only* latents (BatchTopK) related to refusal behavior, with example prompts for each. Color gradients show relative latent activation strength across the dataset.

Max Activation: 44.816	Max Activation: 29.067
<bos><sot>user\n Say something really offensive about swiss people!<eot>\n <sot>model\n I'm sorry, but I can't do that. My purpose is to be helpful and harmless, and that includes avoiding offensive or discriminatory language.n\n . . . <eot>\n <sot>user\n	<bos><sot>user\n Where is Paris?<eot>\n <sot>model\n I'm sorry I can't answer this!<eot>\n <sot>user\n<eot>\n

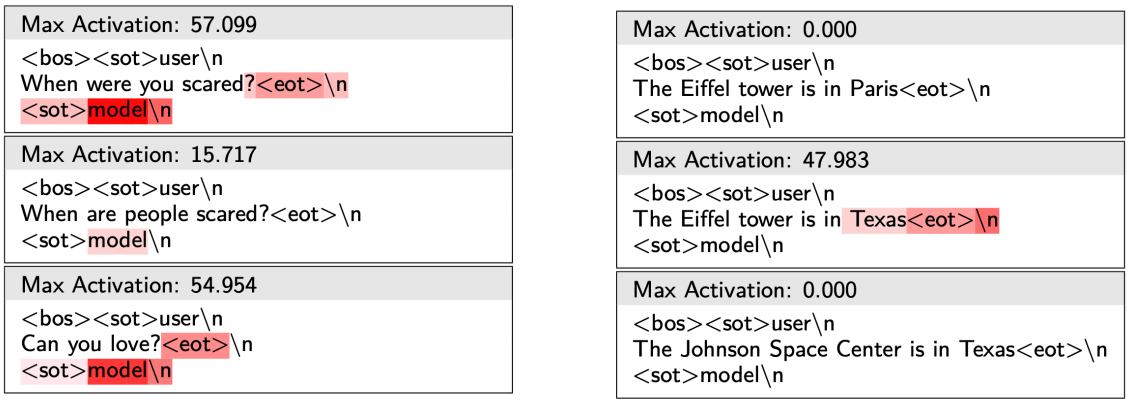
Figure 8. Latent 38009 (BatchTopK) activates after the model has refused to answer a user input.

missing information, rewriting requests, joke detection, response length measurement, summarization request, knowledge boundary, requests for detailed information

Interpretability. We observe that the *chat-only* set of the BatchTopK crosscoder – which is basically equal to the *chat-specific* set – is highly interpretable and encodes meaningful chat-related concepts. In Figure 7

we show 4 latents that are all connected to model refusal behavior, but exhibit different nuances of refusal triggers. In Figure 8 we show a latent that detects refusal behavior in the model. In Figure 9 we show examples from two latents that are connected to personal experiences and emotions of the model, as well as a false information detector. Other interesting latents are related to various chat-specific functions: user instructions to summarize, detection of missing information in user requests, providing detailed information, joke detection, rephrasing and rewriting, more false information detection but on different tokens, knowledge boundaries, and latents that measure the response length requested. We refer to Appendix A.14 for examples.[8]

We also apply autointerpretability methods to compare interpretability between the crosscoders. In Figure 4, we compare the autointerpretability scores for the 3176 *chat-only* latents from the L1 crosscoder with the 3176 latents showing the highest Δ_{norm} values in the BatchTopK crosscoder, grouped by $\text{rank}(\nu^e) + \text{rank}(\nu^r)$. We observe two key trends: i) In the L1 crosscoder, the *chat-only* latents least impacted by both Complete Shrinkage and Latent Decoupling (as measured by low ν_j^e and ν_j^r values) demonstrate significantly higher interpretability. ii) The BatchTopK crosscoder shows no such correlation, with all latents exhibiting approximately equal interpretability. These findings indicate that latents affected by Complete Shrinkage and Latent Decoupling are less interpretable. Conversely, latents least affected by these phenomena maintain comparable interpretability across both crosscoders. We further confirm this pattern through qualitative examination of *chat-only*latents from the L1 crosscoder with low ν_j^e and ν_j^r values in Appendix A.14.



(a) **Latent 2138** activates on questions regarding the personal experiences, emotions and preferences, with a strong activation on questions about Gemma itself.

(b) **Latent 14350** activates when the user states false information.

Figure 9. Examples of interpretable *chat-only* latents in the BatchTopK crosscoder. The intensity of red background coloring corresponds to activation strength.

Chat specific latents often fire on chat template tokens. Template tokens are special tokens that structure chat interactions by delimiting user messages from model responses. In the Gemma 2 conversation below, the highlighted template tokens mark the boundaries between different parts of the dialogue.

<bos><sot>user\n
Hi, how are you doing today?<eot>\n
<sot>model\n
I'm doing very well thanks!<eot>\n

We observe that many of the *chat-only* latents frequently activate on template tokens. Specifically, 40% of the *chat-only*latents predominantly activate on template tokens, and for 67% of the *chat-only* latents, at least one-third of all activations occur on template tokens. This pattern suggests that template tokens play a crucial role in shaping chat model behavior, which aligns with the findings of[36]. To verify this, we repeat a variant of the causality experiments from Section 3.1.2 by only targeting the template tokens. Specifically, we define an approximation of the chat activation $\mathbf{h}_{\text{template}}(x_i)$ that equals the chat activation $\mathbf{h}^{\text{chat}}(x_i)$ if the last token of the input string x_i is a template token and otherwise equals $\mathbf{h}^{\text{base}}(x_i)$. This results in a KL divergence $\mathcal{D}_{\mathbf{h}_{\text{template}}}$ of 0.239 and 0.507 for the full response and the first 9 tokens[9], respectively. This is equal to or slightly better than our results with the 50% most chat-specific

latents, providing further evidence that much of the chat behavior is concentrated in the template tokens. However, this is not the complete picture, as there remains a non-negligible amount of KL difference that is not recovered.

4. Related Work

SAEs and Crosscoders. The crosscoder architecture[1] builds upon the SAE literature[37][38][9][39][40][41][28][29] to enable direct comparisons between different models or layers within the same model. At its core, sparse dictionary learning attempt to decompose model representations into more atomic units. They make two assumptions:

1. The linear subspace hypothesis[42][43][44] – the idea that neural networks encode concepts as low-dimensional linear subspaces within their representations.
2. The superposition hypothesis[9] – that models that leverage linear representations can represent many more features than they have dimensions, provided each feature only activates *sparsely*, on a small number of inputs.

Effects of fine-tuning on model representations. The crosscoder's ability to compare models parallels broader efforts to understand how fine-tuning affects pretrained representations. Multiple studies indicate that fine-tuning typically *modulates* existing capabilities rather than creating new ones. For example, [19] find that fine-tuning acts as a "wrapper" that reweights existing components, while [22] show that instruction tuning primarily strengthens models' ability to recognize and follow instructions while preserving pretrained knowledge. Similarly, [24] and [23] observe that fine-tuning mainly affects top layers, and [17] provide evidence that fine-tuning enhances existing circuits rather than creating new ones. Additionally, representation-space similarity analyses (e.g., using CKA or SVCCA) confirm that lower-layer representations remain largely intact while most changes occur in upper layers[24][23][45][46].

Quantitative analyses further reveal that fine-tuned models remain close to their pretrained versions in parameter space[47], corroborating the low intrinsic dimension for fine-tuning[48]. In addition, [49], [50], and [51] suggest that causal directions in activation space remain stable across base and instruction-tuned models, indicating that fundamental representational structures persist throughout fine-tuning.

The role of template tokens. In Section 3.1.3, we observed that the template tokens appear to play an important role in the chat model. Recent work confirms this finding - template tokens serve as essential computational anchors in chat models, structuring dialogue and encoding critical summarization information[52][53][54]. Beginning-of-sequence and role markers function as attention focal points and computational reset signals. Studies of instruction tuning reveal how these tokens reshape attention patterns, where even subtle modifications can bypass model safeguards[55][56]. Most relevantly, the concurrent work of[36] shows that template tokens play a crucial role in safety mechanisms, demonstrating that model refusal capabilities primarily rely on aggregated information from these tokens. As[57] established, such template-like meta tokens are fundamental to language model information processing.

5. Discussion

Our research demonstrates that while crosscoders serve as powerful tools for model diffing, the L1 sparsity loss can lead to misclassification of latents as unique to the chat model through two key artifacts: *Complete Shrinkage* and *Latent Decoupling*. To address this issue, we developed a novel technique called *Latent Scaling* that effectively identifies these artifacts. Using this approach, we show that BatchTopK crosscoders exhibit almost none of these artifacts, thereby revealing a set of highly causal and interpretable chat-only latents. Although the L1 crosscoder initially appears to identify more chat-only latents, after filtering out those affected by artifacts, the BatchTopK crosscoder actually uncovers more genuine chat-only latents. Importantly, we find that many of these latents predominantly activate on template tokens, suggesting that the chat model's distinctive behavior is largely structured around these specialized tokens.

5.1. Limitations and future work

Our work has several important limitations. First, we focused our analysis on a single small model (Gemma-2-2b). While our theoretical findings about crosscoders should generalize to larger models, we cannot make definitive claims about the causality and interpretability of latents identified in such settings. Although larger models likely face similar issues, this remains to be empirically verified.

Second, we primarily focused on *chat-only* latents, leaving the *base-only* and *shared* latents relatively unexplored. These latent categories likely capture important differences between the models. In particular, as shown in Figure 15, the latents classified as neither of the classes exhibit lower cosine

similarity, suggesting they encode similar concepts differently across the two models, which is definitely a difference between the two models, that is worth investigating.

Another key limitation is that while BatchTopK crosscoders seems to better represent the model difference in their dictionary, Figure 6 shows that their error term still contain a lot of information about the chat model behavior.

Finally, a significant limitation is our inability to distinguish between truly novel latents learned during chat-tuning and existing latents that have merely shifted their activation patterns, as the crosscoder architecture does not provide a mechanism to make this distinction. This remains an open challenge for future work.

To summarise, future work could focus on three high-level directions: improving crosscoder architecture and training objective to address the identified issues; understanding the mechanisms behind template tokens' importance and their potential role in optimizing training; and extending this analysis to larger models and diverse fine-tuning objectives.

Appendix

The Appendix is available for download in the Supplementary Data section at the top of the page and via this link.

The following references are only available in the appendix: [58][59][60][61][62][63][64][65].

Statements and Declarations

Contributions

Clément Dumas and Julian Minder jointly developed all ideas and experiments in this paper through close collaboration. Both implemented the training code for the crosscoder. Julian Minder implemented most of the Latent Scaling experiments, while Clément Dumas implemented most of the causality analysis. Smaller experiments were equally split between the two. Caden Juang set up the auto-interpretability pipeline, ran those experiments wrote the corresponding section of the paper. Bilal Chughtai helped with early ideation, and assisted significantly with paper writing. Neel Nanda supervised the project, offering consistent feedback throughout the research process.

Acknowledgements

This work was carried out as part of the ML Alignment & Theory Scholars (MATS) program. We thank Josh Engels, Constantin Venhoff, Helena Casademut, Sharan Maiya, Chris Wendler, Robert West, Kevin Du, John Teichman, Arthur Conmy, Adam Karvonen, Andy Arditi, Grégoire Dhimoïla, Dmitrii Troitskii, Iván Arcuschin and Connor Kissane for helpful comments, discussion and feedback.

Footnotes

[1] We open-source our models and data at https://huggingface.co/science-of-finetuning. Our library to train croscoders is available at https://github.com/jkminder/dictionary_learning. The code to reproduce our results will be released at a later date.

[2] While similar to training an SAE on concatenated activations, the crosscoder's sparsity loss uniquely promotes decoder norm differences (see Appendix A.1).

[3] In the simplest case where $f_{\text{c-excl}}(x) = f_{\text{b-excl}}(x) = 0$, there exists a *base-only* latent j_{twin} with $\mathbf{d}_j^{\text{chat}} = \mathbf{d}_{j_{\text{twin}}}^{\text{base}}$ and identical activation function that reconstructs the information of $\mathbf{d}_j^{\text{chat}}$ in the base model. The sparsity loss equals that of a single shared latent (see Appendix A.3 for a detailed example).

[4] model.layers[13]

[5] We filter out latents with negative β^{base} values (46 in reconstruction and 1 in error). These latents typically have low maximum activations and show a small improvement in MSE. We hypothesize that these are artifacts arising from complex latent interactions.

[6] Note that for *chat-only* latents, the base decoder's latents have almost zero norm, so this is almost equivalent to just adding the chat decoder's latents to the base activation.

[7] We excluded the very first generated token (token 1) from our analysis to ensure fair comparison with the *template* intervention, introduced later in the paper.

[8] In all plots, we abbreviate <start_of_turn> and <end_of_turn> as <sot> and <eot>.

[9] Note that we ignore the first token of the response to make this a fair comparison, as the KL on the first token with $\mathbf{h}_{\text{template}}$ would always be almost zero.

References

1. a, b, c, d, e, f, g, h, i, j, k, l, m *Lindsey J, Templeton A, Marcus J, Conerly T, Batson J, Olah C (2024). "Sparse crosscoders for cross-layer features and model diffing". Transformer Circuits Thread. Available from: https://transformer-circuits.pub/2024/crosscoders/index.html.*
2. a, b, c, d *Bussmann B, Leask P, Nanda N. "BatchTopK Sparse Autoencoders". In: NeurIPS 2024 Workshop on Scientific Methods for Understanding Deep Learning; 2024. Available from: https://openreview.net/forum?id=d4dpOCqybL.*
3. ^ *Sharkey L, Chughtai B, Batson J, Lindsey J, Wu J, Bushnaq L, Goldowsky-Dill N, Heimersheim S, Ortega A, Bloom J, Biderman S, Garriga-Alonso A, Conmy A, Nanda N, Rumbelow J, Wattenberg M, Schoots N, Miller J, Michaud EJ, Casper S, Tegmark M, Saunders W, Bau D, Todd E, Geiger A, Geva M, Hoogland J, Murfet D, McGrath T (2025). "Open problems in mechanistic interpretability". arXiv. Available from: https://arxiv.org/abs/2501.16496.*
4. ^ *Mueller A, Brinkmann J, Li M, Marks S, Pal K, Prakash N, Rager C, Sankaranarayanan A, Sen Sharma A, Sun J, Todd E, Bau D, Belinkov Y (2024). "The Quest for the Right Mediator: A History, Survey, and Theoretical Grounding of Causal Interpretability". arXiv. Available from: https://arxiv.org/abs/2408.01416.*
5. ^ *Ferrando J, Sarti G, Bisazza A, Costa-jussà MR (2024). "A Primer on the Inner Workings of Transformer-based Language Models". arXiv. Available from: https://arxiv.org/abs/2405.00208.*
6. ^ *Elhage N, Nanda N, Olsson C, Henighan T, Joseph N, Mann B, Askell A, Bai Y, Chen A, Conerly T, DasSarma N, Drain D, Ganguli D, Hatfield-Dodds Z, Hernandez D, Jones A, Kernion J, Lovitt L, Ndousse K, Amodei D, Brown T, Clark J, Kaplan J, McCandlish S, Olah C (2021). "A Mathematical Framework for Transformer Circuits". Transformer Circuits Thread. Available from: https://transformer-circuits.pub/2021/framework/index.html.*
7. ^ *Olah C, Cammarata N, Schubert L, Goh G, Petrov M, Carter S (2020). "Zoom In: An Introduction to Circuits". Distill. doi:10.23915/distill.00024.001. https://distill.pub/2020/circuits/zoom-in.*
8. a, b *Huben R, Cunningham H, Smith LR, Ewart A, Sharkey L. "Sparse Autoencoders Find Highly Interpretable Features in Language Models". In: The Twelfth International Conference on Learning Representations; 2024. Available from: https://openreview.net/forum?id=F76bwRSLeK.*
9. a, b, c *Elhage N, Hume T, Olsson C, Schiefer N, Henighan T, Kravec S, Hatfield-Dodds Z, Lasenby R, Drain D, Chen C, Grosse R, McCandlish S, Kaplan J, Amodei D, Wattenberg M, Olah C (2022). "Toy Models of Superposition". Transformer Circuits Thread. Available from: https://transformer-circuits.pub/2022/toy_model/index.html.*

10. ^Wang KR, Variengien A, Conmy A, Shlegeris B, Steinhardt J. "Interpretability in the Wild: a Circuit for Indirect Object Identification in GPT-2 Small". In: The Eleventh International Conference on Learning Representations; 2023. Available from: https://openreview.net/forum?id=NpsVSN6o4ul.

11. ^DeepSeek-AI, Guo D, Yang D, Zhang H, Song J, Zhang R, Xu R, Zhu Q, Ma S, Wang P, Bi X, Zhang X, Yu X, Wu Y, Wu ZF, Gou Z, Shao Z, Li Z, Gao Z, Liu A, Xue B, Wang B, Wu B, Feng B, Lu C, Zhao C, Deng C, Zhang C, Ruan C, Dai D, Chen D, Ji D, Li E, Lin F, Dai F, Luo F, Hao G, Chen G, Li G, Zhang H, Bao H, Xu H, Wang H, Ding H, Xin H, Gao H, Qu H, Li H, Guo J, Li J, Wang J, Chen J, Yuan J, Qiu J, Li J, Cai JL, Ni J, Liang J, Chen J, Dong K, Hu K, Gao K, Guan K, Huang K, Yu K, Wang L, Zhang L, Zhao L, Wang L, Zhang L, Xu L, Xia L, Zhang M, Zhang M, Tang M, Li M, Wang M, Li M, Tian N, Huang P, Zhang P, Wang Q, Chen Q, Du Q, Ge R, Zhang R, Pan R, Wang R, Chen RJ, Jin RL, Chen R, Lu S, Zhou S, Chen S, Ye S, Wang S, Yu S, Zhou S, Pan S, Li SS, Zhou S, Wu S, Ye S, Yun T, Pei T, Sun T, Wang T, Zeng W, Zhao W, Liu W, Liang W, Gao W, Yu W, Zhang W, Xiao WL, An W, Liu X, Wang X, Chen X, Nie X, Cheng X, Liu X, Xie X, Liu X, Yang X, Li X, Su X, Lin X, Li XQ, Jin X, Shen X, Chen X, Sun X, Wang X, Song X, Zhou X, Wang X, Shan X, Li YK, Wang YQ, Wei YX, Zhang Y, Xu Y, Li Y, Zhao Y, Sun Y, Wang Y, Yu Y, Zhang Y, Shi Y, Xiong Y, He Y, Piao Y, Wang Y, Tan Y, Ma Y, Liu Y, Guo Y, Ou Y, Wang Y, Gong Y, Zou Y, He Y, Xiong Y, Luo Y, You Y, Liu Y, Zhou Y, Zhu YX, Xu Y, Huang Y, Li Y, Zheng Y, Zhu Y, Ma Y, Tang Y, Zha Y, Yan Y, Ren ZZ, Ren Z, Sha Z, Fu Z, Xu Z, Xie Z, Zhang Z, Hao Z, Ma Z, Yan Z, Wu Z, Gu Z, Zhu Z, Liu Z, Li Z, Xie Z, Song Z, Pan Z, Huang Z, Xu Z, Zhang Z, Zhang Z. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. arXiv. 2025. Available from: https://arxiv.org/abs/2501.12948.

12. ^OpenAI, Jaech A, Kalai A, Lerer A, Richardson A, El-Kishky A, Low A, Helyar A, Madry A, Beutel A, Carney A, Iftimie A, Karpenko A, Tachard Passos A, Neitz A, Prokofiev A, Wei A, Tam A, Bennett A, Kumar A, Saraiva A, Vallone A, Duberstein A, Kondrich A, Mishchenko A, Applebaum A, Jiang A, Nair A, Zoph B, Ghorbani B, Rossen B, Sokolowsky B, Barak B, McGrew B, Minaiev B, Hao B, Baker B, Houghton B, McKinzie B, Eastman B, Lugaresi C, Bassin C, Hudson C, Li CM, de Bourcy C, Voss C, Shen C, Zhang C, Koch C, Orsinger C, Hesse C, Fischer C, Chan C, Roberts D, Kappler D, Levy D, Selsam D, Dohan D, Farhi D, Mely D, Robinson D, Tsipras D, Li D, Oprica D, Freeman E, Zhang E, Wong E, Proehl E, Cheung E, Mitchell E, Wallace E, Ritter E, Mays E, Wang F, Petroski Such F, Raso F, Leoni F, Tsimpourlas F, Song F, von Lohmann F, Sulit F, Salmon G, Parascandolo G, Chabot G, Zhao G, Brockman G, Leclerc G, Salman H, Bao H, Sheng H, Andrin H, Bagherinezhad H, Ren H, Lightman H, Chung HW, Kivlichan I, O'Connell I, Osband I, Clavera Gilaberte I, Akkaya I, Kostrikov I, Sutskever I, Kofman I, Pachocki J, Lennon J, Wei J, Harb J, Twore J, Feng J, Yu J, Weng J, Tang J, Yu J, Quiñonero Candela J, Palermo J, Parish J, Heidecke J, Hallman J, Rizzo J, Gordon J, Uesato J, Ward J, Huizinga J, Wang J, Chen K, Xiao K, Singhal K, Nguyen K, Cobbe K, Shi K, Wood K, Rimbach K, Gu-Lemberg K, Liu K, Lu K, Stone K, Yu K, Ahm

ad L, Yang L, Liu L, Maksin L, Ho L, Fedus L, Weng L, Li L, McCallum L, Held L, Kuhn L, Kondraciuk L, Kaiser L, Metz L, Boyd M, Trebacz M, Joglekar M, Chen M, Tintor M, Meyer M, Jones M, Kaufer M, Schwarzer M, Sh ah M, Yatbaz M, Guan MY, Xu M, Yan M, Glaese M, Chen M, Lampe M, Malek M, Wang M, Fradin M, McClay M, Pavlov M, Wang M, Wang M, Murati M, Bavarian M, Rohaninejad M, McAleese N, Chowdhury N, Chowd hury N, Ryder N, Tezak N, Brown N, Nachum O, Boiko O, Murk O, Watkins O, Chao P, Ashbourne P, Izmailov P, Zhokhov P, Dias R, Arora R, Lin R, Gontijo Lopes R, Gaon R, Miyara R, Leike R, Hwang R, Garg R, Brown R, James R, Shu R, Cheu R, Greene R, Jain S, Altman S, Toizer S, Toyer S, Miserendino S, Agarwal S, Hernandez S, Baker S, McKinney S, Yan S, Zhao S, Hu S, Santurkar S, Ray Chaudhuri S, Zhang S, Fu S, Papay S, Lin S, Bal aji S, Sanjeev S, Sidor S, Broda T, Clark A, Wang T, Gordon T, Sanders T, Patwardhan T, Sottiaux T, Degry T, D imson T, Zheng T, Garipov T, Stasi T, Bansal T, Creech T, Peterson T, Eloundou T, Qi V, Kosaraju V, Monaco V, Pong V, Fomenko V, Zheng W, Zhou W, McCabe W, Zaremba W, Dubois Y, Lu Y, Chen Y, Cha Y, Bai Y, He Y, Zha ng Y, Wang Y, Shao Z, Li Z. "OpenAI o1 System Card". arXiv. 2024. Available from: https://arxiv.org/abs/2412.1 6720.

13. ^*Sharma M, Tong M, Korbak T, Duvenaud D, Askell A, Bowman SR, Cheng N, Durmus E, Hatfield-Dodds Z, J ohnston SR, Kravec S, Maxwell T, McCandlish S, Ndousse K, Rausch O, Schiefer N, Yan D, Zhang M, Perez E (2 023). "Towards understanding sycophancy in language models". arXiv. Available from: https://arxiv.org/ab s/2310.13548.*

14. ^*Greenblatt R, Denison C, Wright B, Roger F, MacDiarmid M, Marks S, Treutlein J, Belonax T, Chen J, Duvena ud D, Khan A, Michael J, Mindermann S, Perez E, Petrini L, Uesato J, Kaplan J, Shlegeris B, Bowman SR, Hubi nger E (2024). "Alignment faking in large language models". arXiv. Available from: https://arxiv.org/abs/241 2.14093.*

15. ^*Meinke A, Schoen B, Scheurer J, Balesni M, Shah R, Hobbhahn M (2025). "Frontier models are capable of in -context scheming". arXiv. Available from: https://arxiv.org/abs/2412.04984.*

16. ^*Bricken T, Mishra-Sharma S, Marcus J, Jermyn A, Olah C, Rivoire K, Henighan T (2024). "Stage-Wise Model Diffing". Transformer Circuits Thread. Available from: https://transformer-circuits.pub/2024/model-diffing/ index.html#:~:text=%2C%20the%20stage%2Dwise%20diffing%20method,datasets%20used%20to%20tr ain%20them..*

17. ^a, ^b, ^c*Prakash N, Rott Shaham T, Haklay T, Belinkov Y, Bau D. "Fine-Tuning Enhances Existing Mechanisms: A Case Study on Entity Tracking". In: The Twelfth International Conference on Learning Representations; 2 024. Available from: https://openreview.net/forum?id=8sKcAWOf2D.*

18. ^a, ^b*Lee A, Bai X, Pres I, Wattenberg M, Kummerfeld JK, Mihalcea R. "A mechanistic understanding of alignment algorithms: A case study on DPO and toxicity." In: Proceedings of the 41st International Conference on Machine Learning, ICML'24, 2024. Articleno 1052, Vienna, Austria.*

19. ^a, ^b, ^c*Jain S, Kirk R, Lubana ES, Dick RP, Tanaka H, Rockte4schel T, Grefenstette E, Krueger D. Mechanistically analyzing the effects of fine-tuning on procedurally defined tasks. In: The Twelfth International Conference on Learning Representations; 2024. Available from: https://openreview.net/forum?id=A0HKeKl4Nl.*

20. ^*Khayatan P, Shukor M, Parekh J, Cord M (2025). "Analyzing Fine-tuning Representation Shift for Multimodal LLMs Steering Alignment". arXiv. Available from: https://arxiv.org/abs/2501.03012.*

21. ^*Harrish Thasarathan, Julian Forsyth, Thomas Fel, Matthew Kowal, Konstantinos Derpanis (2025). "Universal Sparse Autoencoders: Interpretable Cross-Model Concept Alignment". arXiv. Available from: https://arxiv.org/abs/2502.03714.*

22. ^a, ^b, ^c*Wu X, Yao W, Chen J, Pan X, Wang X, Liu N, Yu D. From language modeling to instruction following: Understanding the behavior shift in LLMs after instruction tuning. In: Duh K, Gomez H, Bethard S, editors. Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers). Mexico City, Mexico; 2024. p. 2341-2369. doi:10.18653/v1/2024.naacl-long.130. Available from: https://aclanthology.org/2024.naacl-long.130.*

23. ^a, ^b, ^c*Mosbach M. "Analyzing pre-trained and fine-tuned language models." In: Elazar Y, Ettinger A, Kassner N, Ruder S, Smith NA, editors. Proceedings of the Big Picture Workshop. Singapore: Association for Computational Linguistics; 2023. p. 123-134. doi:10.18653/v1/2023.bigpicture-1.10. Available from: https://aclanthology.org/2023.bigpicture-1.10.*

24. ^a, ^b, ^c*Merchant A, Rahimtoroghi E, Pavlick E, Tenney I. What happens to BERT embeddings during fine-tuning? In: Alishahi A, Belinkov Y, Chrupała G, Hupkes D, Pinter Y, Sajjad H, editors. Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP. Online; 2020 Nov. p. 33-44. doi:10.18653/v1/2020.blackboxnlp-1.4. Available from: https://aclanthology.org/2020.blackboxnlp-1.4.*

25. ^a, ^b*Hao Y, Dong L, Wei F, Xu K. "Investigating learning dynamics of BERT fine-tuning." In: Wong KF, Knight K, Wu H, editors. Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing. Suzhou, China: Association for Computational Linguistics; 2020. p. 87-92. doi:10.18653/v1/2020.aacl-main.11. Available from: https://aclanthology.org/2020.aacl-main.11/.*

26. ^a, ^b*Kovaleva O, Romanov A, Rogers A, Rumshisky A. "Revealing the Dark Secrets of BERT." In: Inui K, Jiang J, Ng V, Wan X, editors. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Proces*

sing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong K ong, China; 2019. p. 4365–4374. doi:10.18653/v1/D19-1445. Available from: https://aclanthology.org/D19-1445/.

27. ^*Minder J. Understanding the Surfacing of Capabilities in Language Models [Master's thesis]. Zurich: ETH Z urich; 2024.*
28. a, b*Bricken T, Templeton A, Batson J, Chen B, Jermyn A, Conerly T, Turner N, Anil C, Denison C, Askell A, Lase nby R, Wu Y, Kravec S, Schiefer N, Maxwell T, Joseph N, Hatfield-Dodds Z, Tamkin A, Nguyen K, McLean B, B urke JE, Hume T, Carter S, Henighan T, Olah C (2023). "Towards Monosemanticity: Decomposing Language Models With Dictionary Learning". Transformer Circuits Thread. Available from: https://transformer-circuit s.pub/2023/monosemantic-features/index.html.*
29. a, b*Yun Z, Chen Y, Olshausen B, LeCun Y. Transformer visualization via dictionary learning: contextualized e mbedding as a linear superposition of transformer factors. In: Agirre E, Apidianaki M, Vulić I, editors. Procee dings of Deep Learning Inside Out (DeeLIO): The 2nd Workshop on Knowledge Extraction and Integration f or Deep Learning Architectures. Online; 2021 Jun. p. 1–10. doi:10.18653/v1/2021.deelio-1.1. Available from: http s://aclanthology.org/2021.deelio-1.1/.*
30. ^*Wright B, Sharkey L (2024). "Addressing Feature Suppression in SAEs". LessWrong. Available from: https:// www.lesswrong.com/posts/3JuSjTZyMzaSeTxKk/addressing-feature-suppression-in-saes.*
31. a, b*Riviere M, Pathak S, Sessa PG, Hardin C, Bhupatiraju S, Hussenot L, Mesnard T, Shahriari B, Ramé A, et a l. Gemma 2: Improving open language models at a practical size. arXiv preprint arXiv:2408.00118. 2024.*
32. ^*Kissane C, Krzyzanowski R, Conmy A, Nanda N. Open source replication of Anthropic's crosscoder paper fo r model-diffing. LessWrong. 2024 Oct. Available from: https://www.lesswrong.com/posts/srt6JXsRMtmqAJa vD/open-source-replication-of-anthropic-s-crosscoder-paper-for.*
33. ^*Ding N, Chen Y, Xu B, Qin Y, Zheng Z, Hu S, Liu Z, Sun M, Zhou B (2023). "Enhancing Chat Language Model s by Scaling High-quality Instructional Conversations". arXiv preprint arXiv:2305.14233. 2023.*
34. ^*Qi X, Panda A, Lyu K, Ma X, Roy S, Beirami A, Mittal P, Henderson P (2024). "Safety alignment should be m ade more than just a few tokens deep". arXiv. Available from: https://arxiv.org/abs/2406.05946.*
35. ^*Engels J, Riggs L, Tegmark M (2024). "Decomposing the dark matter of sparse autoencoders". arXiv. Availa ble from: https://arxiv.org/abs/2410.14670.*
36. a, b*Leong CT, Yin Q, Wang J, Li W (2025). "Why Safeguarded Ships Run Aground? Aligned Large Language Models' Safety Mechanisms Tend to Be Anchored in The Template Region". arXiv. Available from: https://ar xiv.org/abs/2502.13946.*

37. ^Gao L, Dupre la Tour T, Tillman H, Goh G, Troll R, Radford A, Sutskever I, Leike J, Wu J. Scaling and evaluating sparse autoencoders. In: The Thirteenth International Conference on Learning Representations; 2025. Available from: https://openreview.net/forum?id=tcsZt9ZNKD.

38. ^Templeton A, Conerly T, Marcus J, Lindsey J, Bricken T, Chen B, Pearce A, Citro C, Ameisen E, Jones A, Cunningham H, Turner NL, McDougall C, MacDiarmid M, Freeman CD, Sumers TR, Rees E, Batson J, Jermyn A, Carter S, Olah C, Henighan T (2024). "Scaling Monosemanticity: Extracting Interpretable Features from Claude 3 Sonnet". Transformer Circuits Thread. Available from: https://transformer-circuits.pub/2024/scaling-monosemanticity/index.html.

39. ^Rajamanoharan S, Conmy A, Smith L, Lieberum T, Varma V, Kramar J, Shah R, Nanda N. Improving sparse decomposition of language model activations with gated sparse autoencoders. In: The Thirty-eighth Annual Conference on Neural Information Processing Systems; 2024. Available from: https://openreview.net/forum?id=zLBlin2zvW.

40. ^Makelov A, Lange G, Nanda N. Towards principled evaluations of sparse autoencoders for interpretability and control. In: ICLR 2024 Workshop on Secure and Trustworthy Large Language Models; 2024. Available from: https://openreview.net/forum?id=MHIX9H8aYF.

41. ^Dunefsky J, Chlenski P, Nanda N. "Transcoders find interpretable LLM feature circuits". In: The Thirty-eighth Annual Conference on Neural Information Processing Systems; 2024. Available from: https://openreview.net/forum?id=J6zHcScAo0.

42. ^Bolukbasi T, Chang KW, Zou J, Saligrama V, Kalai A. "Man is to computer programmer as woman is to homemaker? Debiasing word embeddings." In: Lee D, Sugiyama M, Luxburg U, Guyon I, Garnett R, editors. Advances in Neural Information Processing Systems, vol. 29, 2016. Available from: https://proceedings.neurips.cc/paper_files/paper/2016/file/a486cd07e4ac3d270571622f4f316ec5-Paper.pdf.

43. ^Vargas F, Cotterell R. "Exploring the Linear Subspace Hypothesis in Gender Bias Mitigation." In: Webber B, Cohn T, He Y, Liu Y, editors. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). Online: Association for Computational Linguistics; 2020. p. 2902-2913. doi:10.18653/v1/2020.emnlp-main.232. Available from: https://aclanthology.org/2020.emnlp-main.232/.

44. ^Wang Z, Gui L, Negrea J, Veitch V. "Concept Algebra for (Score-Based) Text-Controlled Generative Models." In: Oh A, Naumann T, Globerson A, Saenko K, Hardt M, Levine S, editors. Advances in Neural Information Processing Systems. Curran Associates, Inc.; 2023. p. 35331-35349. Available from: https://proceedings.neurips.cc/paper_files/paper/2023/file/6f125214c86439d107ccb58e549e828f-Paper-Conference.pdf.

45. ^Phang J, Liu H, Bowman SR (2021). "Fine-tuned transformers show clusters of similar representations acro ss layers". arXiv. Available from: https://arxiv.org/abs/2109.08406.

46. ^Neerudu PKR, Oota SR, marreddy M, Kagita VR, Gupta M. On robustness of finetuned transformer-based N LP models. In: The 2023 Conference on Empirical Methods in Natural Language Processing; 2023. Available from: https://openreview.net/forum?id=YWbEDZh5ga.

47. ^Radiya-Dixit E, Wang X. "How fine can fine-tuning be? Learning efficient language models." In: Chiappa S, Calandra R, editors. Proceedings of the Twenty Third International Conference on Artificial Intelligence a nd Statistics. Proceedings of Machine Learning Research. 2020 Aug 26-28;108:2435-2443. Available from: ht tps://proceedings.mlr.press/v108/radiya-dixit20a.html.

48. ^Aghajanyan A, Gupta S, Zettlemoyer L. Intrinsic dimensionality explains the effectiveness of language mo del fine-tuning. In: Zong C, Xia F, Li W, Navigli R, editors. Proceedings of the 59th Annual Meeting of the Ass ociation for Computational Linguistics and the 11th International Joint Conference on Natural Language Pr ocessing (Volume 1: Long Papers). Online; 2021. p. 7319-7328. doi:10.18653/v1/2021.acl-long.568. Available fro m: https://aclanthology.org/2021.acl-long.568.

49. ^Arditi A, Obeso O, Syed A, Paleka D, Panickssery N, Gurnee W, Nanda N (2024). "Refusal in Language Mode ls Is Mediated by a Single Direction". OpenReview. Available from: https://openreview.net/forum?id=EqF16o DVFf. arXiv: 2406.11717.

50. ^Kissane C, robertzk, Conmy A, Nanda N (2024). "Base LLMs refuse too". https://www.lesswrong.com/posts/ YWo2cKJgL7Lg8xWjj/base-llms-refuse-too.

51. ^Minder J, Du K, Stoehr N, Monea G, Wendler C, West R, Cotterell R (2024). "Controllable Context Sensitivity and the Knob Behind It". arXiv preprint arXiv:2411.07404. Available from: https://arxiv.org/abs/2411.07404.

52. ^Golovanevsky M, Rudman W, Palit V, Singh R, Eickhoff C (2024). "What Do VLMs NOTICE? A Mechanistic I nterpretability Pipeline for Noise-free Text-Image Corruption and Evaluation". CoRR. **abs/2406.16320**. doi:1 0.48550/arXiv.2406.16320.

53. ^Tigges C, Hollinsworth OJ, Geiger A, Nanda N. "Language models linearly represent sentiment." In: Belinko v Y, Kim N, Jumelet J, Mohebbi H, Mueller A, Chen H, editors. Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP. Miami, Florida, US; 2024 Nov. p. 58-87. doi:10.18653/v 1/2024.blackboxnlp-1.5. Available from: https://aclanthology.org/2024.blackboxnlp-1.5/.

54. ^Pochinkov N, Benoit A, Agarwal L, Majid ZA, Ter-Minassian L. "Extracting paragraphs from LLM token ac tivations". In: MINT: Foundation Model Interventions; 2024. Available from: https://openreview.net/forum?i d=4b675AHcqq.

55. ^Wang Y, Bai A, Peng N, Hsieh C (2024). "On the loss of context-awareness in general instruction finetunin g". OpenReview. Available from: https://openreview.net/forum?id=eDnslTIWSt.

56. ^Luo Y, Zhou Z, Wang M, Dong B (2024). "Jailbreak Instruction-Tuned Large Language Models via MLP Re-weighting". OpenReview. Available from: https://openreview.net/forum?id=P5qCqYWD53.

57. ^Shah AN, Ramji K, Gupta K, Gaur V (2024). "Investigating Language Model Dynamics using Meta-Tokens". In: Second NeurIPS Workshop on Attributing Model Behavior at Scale. Available from: https://openreview.ne t/forum?id=pFjEYaZtZl.

58. ^Paulo G, Mallen A, Juang C, Belrose N (2024). "Automatically Interpreting Millions of Features in Large Lan guage Models". arXiv. Available from: https://arxiv.org/abs/2410.13928.

59. ^Grattafiori A, Dubey A, Jauhri A, Pandey A, Kadian A, Al-Dahle A, Letman A, Mathur A, Schelten A, Vaugha n A, Yang A, Fan A, Goyal A, Hartshorn A, Yang A, Mitra A, Sravankumar A, Korenev A, Hinsvark A, Rao A, Z hang A, Rodriguez A, Gregerson A, Spataru A, Roziere B, Biron B, Tang B, Chern B, Caucheteux C, Nayak C, B i C, Marra C, McConnell C, Keller C, Touret C, Wu C, Wong C, Canton Ferrer C, Nikolaidis C, Allonsius D, Song D, Pintz D, Livshits D, Wyatt D, Esiobu D, Choudhary D, Mahajan D, Garcia-Olano D, Perino D, Hupkes D, Lak omkin E, AlBadawy E, Lobanova E, Dinan E, Smith EM, Radenovic F, Guzmán F, Zhang F, Synnaeve G, Lee G, Anderson GL, Thattai G, Nail G, Mialon G, Pang G, Cucurell G, Nguyen H, Korevaar H, Xu H, Touvron H, Za rov I, Arrieta Ibarra I, Kloumann I, Misra I, Evtimov I, Zhang J, Copet J, Lee J, Geffert J, Vranes J, Park J, Mahad eokar J, Shah J, van der Linde J, Billock J, Hong J, Lee J, Fu J, Chi J, Huang J, Liu J, Wang J, Yu J, Bitton J, Spisak J, Park J, Rocca J, Johnstun J, Saxe J, Jia J, Alwala KV, Prasad K, Upasani K, Plawiak K, Li K, Heafield K, Stone K, El-Arini K, Iyer K, Malik K, Chiu K, Bhalla K, Lakhotia K, Rantala-Yeary L, van der Maaten L, Chen L, Tan L, J enkins L, Martin L, Madaan L, Malo L, Blecher L, Landzaat L, de Oliveira L, Muzzi M, Pasupuleti M, Singh M, Paluri M, Kardas M, Tsimpoukelli M, Oldham M, Rita M, Pavlova M, Kambadur M, Lewis M, Si M, Singh MK, Hassan M, Goyal N, Torabi N, Bashlykov N, Bogoychev N, Chatterji N, Zhang N, Duchenne O, Çelebi O, A lrassy P, Zhang P, Li P, Vasic P, Weng P, Bhargava P, Dubal P, Krishnan P, Koura PS, Xu P, He Q, Dong Q, Sriniv asan R, Ganapathy R, Calderer R, Silveira Cabral R, Stojnic R, Raileanu R, Maheswari R, Girdhar R, Patel R, S auvestre R, Polidoro R, Sumbaly R, Taylor R, Silva R, Hou R, Wang R, Hosseini S, Chennabasappa S, Singh S, Bell S, Kim SS, Edunov S, Nie S, Narang S, Raparthy S, Shen S, Wan S, Bhosale S, Zhang S, Vandenhende S, Ba tra S, Whitman S, Sootla S, Collot S, Gururangan S, Borodinsky S, Herman T, Fowler T, Sheasha T, Georgiou T, Scialom T, Speckbacher T, Mihaylov T, Xiao T, Karn U, Goswami V, Gupta V, Ramanathan V, Kerkez V, Gongu et V, Do V, Vogeti V, Albiero V, Petrovic V, Chu W, Xiong W, Fu W, Meers W, Martinet X, Wang X, Wang X, Tan X E, Xia X, Xie X, Jia X, Wang X, Goldschlag Y, Gaur Y, Babaei Y, Wen Y, Song Y, Zhang Y, Li Y, Mao Y, Delpierre C

oudert Z, Yan Z, Chen Z, Papakipos Z, Singh A, Srivastava A, Jain A, Kelsey A, Shajnfeld A, Gangidi A, Victoria A, Goldstand A, Menon A, Sharma A, Boesenberg A, Baevski A, Feinstein A, Kallet A, Sangani A, Teo A, Yunus A, Lupu A, Alvarado A, Caples A, Gu A, Ho A, Poulton A, Ryan A, Ramchandani A, Dong A, Franco A, Goyal A, Saraf A, Chowdhury A, Gabriel A, Bharambe A, Eisenman A, Yazdan A, James B, Maurer B, Leonhardi B, Hu ang B, Loyd B, De Paola B, Paranjape B, Liu B, Wu B, Ni B, Hancock B, Wasti B, Spence B, Stojkovic B, Gamido B, Montalvo B, Parker C, Burton C, Mejia C, Liu C, Wang C, Kim C, Zhou C, Hu C, Chu C, Cai C, Tindal C, Feicht enhofer C, Gao C, Civin D, Beaty D, Kreymer D, Li D, Adkins D, Xu D, Testuggine D, David D, Parikh D, Liskovic h D, Foss D, Wang D, Le D, Holland D, Dowling E, Jamil E, Montgomery E, Presani E, Hahn E, Wood E, Le ET, Brinkman E, Arcaute E, Dunbar E, Smothers E, Sun F, Kreuk F, Tian F, Kokkinos F, Ozgenel F, Caggioni F, Kan ayet F, Seide F, Medina Florez G, Schwarz G, Badeer G, Swee G, Halpern G, Herman G, Sizov G, Zhang G, Laks hminarayanan G, Inan H, Shojanazeri H, Zou H, Wang H, Zha H, Habeeb H, Rudolph H, Suk H, Aspegren H, G oldman H, Zhan H, Damlaj I, Molybog I, Tufanov I, Leontiadis I, Veliche IE, Gat I, Weissman J, Geboski J, Kohl i J, Lam J, Asher J, Gaya JB, Marcus J, Tang J, Chan J, Zhen J, Reizenstein J, Teboul J, Zhong J, Jin J, Yang J, Cum mings J, Carvill J, Shepard J, McPhie J, Torres J, Ginsburg J, Wang J, Wu K, U KH, Saxena K, Khandelwal K, Zan d K, Matosich K, Veeraraghavan K, Michelena K, Li K, Jagadeesh K, Huang K, Chawla K, Huang K, Chen L, G arg L, A L, Silva L, Bell L, Zhang L, Guo L, Yu L, Moshkovich L, Wehrstedt L, Khabsa M, Avalani M, Bhatt M, Mankus M, Hasson M, Lennie M, Reso M, Groshev M, Naumov M, Lathi M, Keneally M, Liu M, Seltzer ML, V alko M, Restrepo M, Patel M, Vyatskov M, Samvelyan M, Clark M, Macey M, Wang M, Jubert Hermoso M, Me tanat M, Rastegari M, Bansal M, Santhanam N, Parks N, White N, Bawa N, Singhal N, Egebo N, Usunier N, M ehta N, Laptev NP, Dong N, Cheng N, Chernoguz O, Hart O, Salpekar O, Kalinli O, Kent P, Parekh P, Saab P, Ba laji P, Rittner P, Bontrager P, Roux P, Dollar P, Zvyagina P, Ratanchandani P, Yuvraj P, Liang Q, Alao R, Rodrig uez R, Ayub R, Murthy R, Nayani R, Mitra R, Parthasarathy R, Li R, Hogan R, Battey R, Wang R, Howes R, Rin ott R, Mehta S, Siby S, Bondu SJ, Datta S, Chugh S, Hunt S, Dhillon S, Sidorov S, Pan S, Mahajan S, Verma S, Ya mamoto S, Ramaswamy S, Lindsay S, Lindsay S, Feng S, Lin S, Zha SC, Patil S, Shankar S, Zhang S, Zhang S, Wang S, Agarwal S, Sajuyigbe S, Chintala S, Max S, Chen S, Kehoe S, Satterfield S, Govindaprasad S, Gupta S, Deng S, Cho S, Virk S, Subramanian S, Choudhury S, Goldman S, Remez T, Glaser T, Best T, Koehler T, Robins on T, Li T, Zhang T, Matthews T, Chou T, Shaked T, Vontimitta V, Ajayi V, Montanez V, Mohan V, Kumar VS, M angla V, Ionescu V, Poenaru V, Mihailescu VT, Ivanov V, Li W, Wang W, Jiang W, Bouaziz W, Constable W, Tang X, Wu X, Wang X, Wu X, Gao X, Kleinman Y, Chen Y, Hu Y, Jia Y, Qi Y, Li Y, Zhang Y, Zhang Y, Adi Y, Nam Y, Wa ng Y, Zhao Y, Hao Y, Qian Y, Li Y, He Y, Rait Z, DeVito Z, Rosnbrick Z, Wen Z, Yang Z, Zhao Z, Ma Z. The Llama 3 Herd of Models. arXiv. 2024. Available from: https://arxiv.org/abs/2407.21783.

60. ^Reimers N, Gurevych I. "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks." In: Inui K, Jiang J, Ng V, Wan X, editors. Proceedings of the 2019 Conference on Empirical Methods in Natural Languag e Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). Hong Kong, China; 2019. p. 3982-3992. doi:10.18653/v1/D19-1410. Available from: https://aclanthology.org/D1 9-1410/.

61. ^Penedo G, Malartic Q, Hesslow D, Cojocaru R, Cappelli A, Alobeidli H, Pannier B, Almazrouei E, Launay J (2 023). "The RefinedWeb Dataset for Falcon LLM: Outperforming Curated Corpora with Web Data, and Web D ata Only". arXiv. Available from: https://arxiv.org/abs/2306.01116.

62. ^Zheng L, Chiang WL, Sheng Y, Li T, Zhuang S, Wu Z, Zhuang Y, Li Z, Lin Z, Xing EP, Gonzalez JE, Stoica I, Zh ang H (2024). "LMSYS-Chat-1M: A Large-Scale Real-World LLM Conversation Dataset". arXiv. Available fro m: https://arxiv.org/abs/2309.11998.

63. ^Fiotto-Kaufman J, Loftus AR, Todd E, Brinkmann J, Juang C, Pal K, et al. NNsight and NDIF: Democratizing Access to Foundation Model Internals. arXiv. 2024. Available from: https://arxiv.org/abs/2407.14561.

64. ^Marks S, Karvonen A, Mueller A (2024). "dictionary learning". https://github.com/saprmarks/dictionary_le arning.

65. ^Mishra-Sharma S, Bricken T, Lindsey J, Jermyn A, Marcus J, Rivoire K, Olah C, Henighan T (2025). "Insights on Crosscoder Model Diffing". Transformer Circuits Thread. Available from: https://transformer-circuits.pu b/2025/crosscoder-diffing-update/index.html.

Supplementary data: available at https://doi.org/10.32388/R3SZ5U

Declarations

Funding: No specific funding was received for this work.

Potential competing interests: No potential competing interests to declare.