Robustly identifying concepts introduced during chat fine-tuning using crosscoders

Content Warning: This paper contains examples of harmful language

Julian Minder[*,ɘ,ɑ] **Clément Dumas**[*,†,‡]
Bilal Chughtai **Neel Nanda**
[ɘ]EPFL [ɑ]ETHZ [†]Ecole Normale Supérieure Paris-Saclay [‡]Université Paris-Saclay
julian.minder@epfl.ch, clement.dumas@ens-paris-saclay.fr

Abstract

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 (Lindsey et al., 2024) 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 (Bussmann et al., 2024) 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]

1 Introduction

Mechanistic interpretability aims to understand the internal computations of neural networks (Olah et al., 2020; Elhage et al., 2021). A nascent approach, *model diffing*, focuses on identifying what

[*]Equal contribution. Order randomized.

[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.

changes within a model due to fine-tuning. Given the computational constraints of fine-tuning compared to pre-training, changes are expected to be localized, making model diffing a potentially tractable path to understanding task-specific adaptations like instruction following or persona adoption.

Crosscoders (Lindsey et al., 2024), built upon Sparse Autoencoders (SAEs) (Yun et al., 2021; Bricken et al., 2023), offer a promising tool for model diffing. They learn a shared dictionary of concepts ("latents") represented by directions in the activation space of both a base and a fine-tuned (e.g., chat) model. Each latent has a shared activation function but distinct decoder directions ($\mathbf{d}_j^{\text{base}}$, $\mathbf{d}_j^{\text{chat}}$) for reconstructing activations in each model. This allows tracking how concepts are represented across models. Prior work identified "chat-only" latents where the base decoder norm $\|\mathbf{d}_j^{\text{base}}\|_2$ is zero, hypothesizing these represent concepts introduced during fine-tuning.

However, we identify two theoretical artifacts stemming from the standard L1 crosscoder training loss that challenge this interpretation:

1. **Complete Shrinkage:** The L1 sparsity penalty can force $\|\mathbf{d}_j^{\text{base}}\|_2$ to zero even if latent j contributes to base model reconstruction, especially if its contribution is much larger in the chat 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.

These issues can lead to falsely identifying latents as novel to the chat model.

Our contributions are: 1. We theoretically identify and empirically demonstrate these L1 crosscoder issues (Sections 2.3 and 3). 2. We develop *Latent Scaling*, a technique to diagnose these artifacts by measuring a latent's true contribution to each model's error and reconstruction (Section 2.4). 3. We show that crosscoders trained with BatchTopK loss (Bussmann et al., 2024), which enforces sparsity differently, largely avoid these artifacts (Section 3). 4. Using the BatchTopK approach, we identify interpretable and causally relevant chat-specific concepts like nuanced refusals, persona queries, false information detection, and find they often activate strongly on chat template tokens (Section 3).

2 Methods

2.1 Crosscoder Architectures

Given input x, let $\mathbf{h}^{\text{base}}(x), \mathbf{h}^{\text{chat}}(x) \in \mathbb{R}^d$ be activations. For a dictionary of size D, the jth latent activation $f_j(x)$ is:

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

Reconstructions use model-specific decoders $\mathbf{d}_j^{\text{base}}, \mathbf{d}_j^{\text{chat}}$:

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

Let reconstruction errors be $\boldsymbol{\varepsilon}^{\text{base}}(x) = \mathbf{h}^{\text{base}}(x) - \widetilde{\mathbf{h}}^{\text{base}}(x)$ and $\boldsymbol{\varepsilon}^{\text{chat}}(x) = \mathbf{h}^{\text{chat}}(x) - \widetilde{\mathbf{h}}^{\text{chat}}(x)$.

L1 crosscoder. The training loss includes reconstruction error and an L1 sparsity penalty on decoder norms, weighted by activation strength:

$$\mathcal{L}_{\text{L1}}(x) = \frac{1}{2}(\|\boldsymbol{\varepsilon}^{\text{base}}(x)\|_2 + \|\boldsymbol{\varepsilon}^{\text{chat}}(x)\|_2) + \mu \sum_j f_j(x)(\|\mathbf{d}_j^{\text{base}}\|_2 + \|\mathbf{d}_j^{\text{chat}}\|_2) \tag{3}$$

This penalty differs from a standard SAE on concatenated activations (see Appendix A.2 for details).

BatchTopK crosscoder. Sparsity is enforced by selecting only the top $n \cdot k$ latents with the highest scaled activation $v(x_i, j) = f_j(x_i)(\|\mathbf{d}_j^{\text{base}}\|_2 + \|\mathbf{d}_j^{\text{chat}}\|_2)$ across a batch $\mathcal{X}, |\mathcal{X}| = n$. See Appendix A.1 for details.

2.2 Decoder norm based model diffing

Lindsey et al. (2024) classify latents based on the relative norm difference (Δ_{norm}) of their decoders:

$$\Delta_{\text{norm}}(j) = \frac{1}{2}\left(\frac{\|\mathbf{d}_j^{\text{chat}}\|_2 - \|\mathbf{d}_j^{\text{base}}\|_2}{\max\left(\|\mathbf{d}_j^{\text{chat}}\|_2,\ \|\mathbf{d}_j^{\text{base}}\|_2\right)} + 1\right) \tag{4}$$

Values near 1 suggest that a latent is only in the chat model (*chat-only*), near 0 suggest *base-only*, and near 0.5 suggest that the latent is present in both models (*shared*).

2.3 Are L1*chat-only* latents really chat-specific?

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, does $\|\mathbf{d}_j^{\text{base}}\|_2 \approx 0$ imply latent j is truly chat-specific? We identify two reasons (L1 loss artifacts) to doubt this:

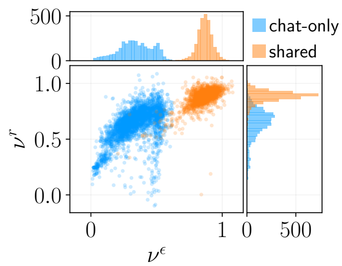
- **Complete Shrinkage.** The L1 regularization term may force the norm of the base decoder vector $\mathbf{d}_j^{\text{base}}$ to be zero, even though it is present in the base activation and could have contributed to the reconstruction of base activation. This may especially be relevant if the contribution of latent j is non-zero in the base model, but much smaller than the contribution in the chat model. Consequently, <u>the error $\boldsymbol{\varepsilon}^{\text{base}}$ contains information that can be attributed to latent j.</u>
- **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, <u>the base reconstruction $\widetilde{\mathbf{h}}^{\text{base}}$ contains information that could be attributed to latent j</u>. See Appendix A.4 for details.

2.4 Latent Scaling: Diagnosing L1 Artifacts

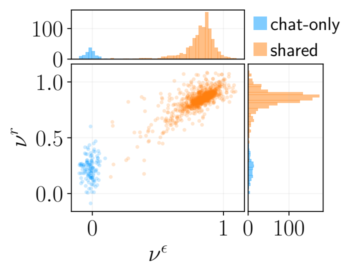
To empirically detect Complete Shrinkage and Latent Decoupling for a given *chat-only* latent j, we measure how well its chat direction $\mathbf{d}_j^{\text{chat}}$, scaled by a factor β, can explain base model activations. We compute two ratios by solving least squares (see Appendix A.5 for the closed form solution):

$$\underset{\beta_j}{\operatorname{argmin}} \sum_{i=0}^{n} \|\beta_j f_j(x_i)\mathbf{d}_j^{\text{chat}} - \mathbf{y}_i^m\|_2^2 \tag{5}$$

1. **Error Ratio (ν_j^ε):** Measures how well $\beta f_j(x)\mathbf{d}_j^{\text{chat}}$ explains the base reconstruction *error* $\boldsymbol{\varepsilon}^{\text{base}}(x)$. High $\nu_j^\varepsilon = \beta_j^{\varepsilon,\text{base}}/\beta_j^{\varepsilon,\text{chat}} \approx 1$ indicates Complete Shrinkage (latent j could explain base error). Truly chat-specific should have $\nu_j^\varepsilon \approx 0$.
2. **Reconstruction Ratio (ν_j^r):** Measures how well $\beta f_j(x)\mathbf{d}_j^{\text{chat}}$ explains the base model's *reconstruction* $\widetilde{\mathbf{h}}^{\text{base}}(x)$. High $\nu_j^r = \beta_j^{r,\text{base}}/\beta_j^{r,\text{chat}} \approx 1$ indicates Latent Decoupling (latent j's information is already present in the base reconstruction, likely via other latents). Truly chat-specific should have low ν_j^r.



(a) L1 crosscoder



(b) BatchTopK crosscoder

Name	Δ_{norm}	Count	
		L1	BTopK
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	62,373

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

Figure 1: ν^ε vs ν^r for *chat-only* latents in L1 (a) and BatchTopK (b). Red points are *shared* latents for reference. High ν^ε (y-axis) suggests Shrinkage; high ν^r (x-axis) overlapping with shared suggests Decoupling. L1 suffers from both, BatchTopK does not. In (c) we show the Δ_{norm} classifications.

Low values for both ν_j^ε and ν_j^r suggest latent j is genuinely chat-specific and robust to these L1 artifacts.

3 Results: Artifacts, Robustness, Interpretability and Causality

We trained L1 and BatchTopK crosscoders on Gemma 2 2B base/chat models (layer 13, expansion 32, L0 sparsity ≈ 100). See Appendix A.12 for full details.

Norm-Based Classification. Figure 1c compares the Δ_{norm} distributions. The L1 crosscoder identifies far more *chat-only* latents (3176) than BatchTopK (134) based on this metric (Figure 1c). However, this includes spurious latents.

Diagnosing Artifacts with Latent Scaling. We applied Latent Scaling to the Δ_{norm}-identified *chat-only* sets. Figure 1 shows the results.

- **L1 Crosscoder:** Many *chat-only* latents exhibit high ν^ε (reaching ≈ 0.5, indicating Complete Shrinkage) and/or high ν^r (significant overlap with the *shared* distribution, indicating Latent Decoupling). 18% overlap with the central 95% of shared ν^r.
- **BatchTopK Crosscoder:** The few *chat-only* latents identified by Δ_{norm} show near-zero ν^ε and low ν^r with no overlap with shared latents, indicating robustness.

Comparing the 3176 L1*chat-only* latents with the top-3176 BatchTopK latents ranked by Δ_{norm}, Figure 8 shows BatchTopK consistently yields far more latents robust to artifacts (low ν^ε and ν^r) at any threshold. We conclude that L1 crosscoders suffer significantly from these artifacts, while BatchTopK crosscoders are robust. The Δ_{norm} metric is unreliable for identifying truly chat-specific latents in L1 models but appears reliable for BatchTopK models.

We also find that the genuinely chat-specific latents are more causal in Appendix A.8.

Interpretation of Chat-Specific Latents. The genuinely chat-specific latents identified (especially using BatchTopK) are highly interpretable. Examples include

- **Refusals:** Multiple latents detecting different triggers for refusal (harmful instructions, sensitive content, unethical requests) or detecting the model's own refusal action (concepts from Figures 13 and 14).

- **Persona & Interaction:** Latents activating on personal questions to the model, queries about its capabilities, or detection of false information provided by the user (concepts from Figure 15).
- **Task Specifics:** Latents related to summarization, joke detection, rewriting requests, knowledge boundaries (examples in Appendix A.16).

A key finding is the role of **template tokens** (e.g., `<sot>`, `<eot>`). We found 40% of BatchTopK *chat-only* latents activate predominantly on template tokens, and 67% activate on them for at least one-third of their occurrences.

4 Conclusion

Crosscoders are valuable for model diffing, but standard L1 implementations suffer from *Complete Shrinkage* and *Latent Decoupling* artifacts, leading to misidentification of chat-specific features. We introduced *Latent Scaling* ($\nu^{\varepsilon}, \nu^{r}$ ratios) to reliably diagnose these issues. We demonstrated that BatchTopK crosscoders, due to their different sparsity mechanism, are largely robust to these artifacts. Causal experiments confirmed that Latent Scaling is necessary to isolate impactful *chat-only* latents in L1 crosscoders, while the simpler Δ_{norm} metric is sufficient for BatchTopK crosscoders. The robustly identified chat-specific latents represent interpretable concepts like nuanced refusals, persona, and false information detection. Crucially, many of these latents, and much of the causal effect of chat tuning, are associated with chat template tokens. For reliable model diffing using crosscoders, we recommend using BatchTopK training or applying diagnostic filtering like Latent Scaling to standard L1 crosscoders. Future work should explore these phenomena in larger models and investigate the mechanisms tied to template tokens further.

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. 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. We would also liked to thanks Caden Juang for his helpful comments and help to run the auto-interpretability pipeline.

References

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. URL `https://transformer-circuits.pub/2023/monosemantic-features/index.html`.

Bart Bussmann, Patrick Leask, and Neel Nanda. Batchtopk sparse autoencoders. In *NeurIPS 2024 Workshop on Scientific Methods for Understanding Deep Learning*, 2024. URL `https://openreview.net/forum?id=d4dpOCqybL`.

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

Jaden Fiotto-Kaufman, Alexander R Loftus, Eric Todd, Jannik Brinkmann, Caden Juang, Koyena Pal, Can Rager, Aaron Mueller, Samuel Marks, Arnab Sen Sharma, Francesca Lucchetti, Michael Ripa, Adam Belfki, Nikhil Prakash, Sumeet Multani, Carla Brodley, Arjun Guha, Jonathan Bell, Byron Wallace, and David Bau. Nnsight and ndif: Democratizing access to foundation model internals. *arXiv*, 2024. URL `https://arxiv.org/abs/2407.14561`.

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. *arXiv*, 2024. URL `https://arxiv.org/abs/2407.21783`.

Connor Kissane, Robert Krzyzanowski, Arthur Conmy, and Neel Nanda. Open source replication of Anthropic's crosscoder paper for model-diffing. *LessWrong*, October 2024. URL `https://www.lesswrong.com/posts/srt6JXsRMtmqAJavD/open-source-replication-of-anthropic-s-crosscoder-paper-for`.

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

Samuel Marks, Adam Karvonen, and Aaron Mueller. dictionary learning. `https://github.com/saprmarks/dictionary_learning`, 2024.

Siddharth Mishra-Sharma, Trenton Bricken, Jack Lindsey, Adam Jermyn, Jonathan Marcus, Kelley Rivoire, Christopher Olah, and Thomas Henighan. Insights on crosscoder model diffing. *Transformer Circuits Thread*, 2025. URL `https://transformer-circuits.pub/2025/crosscoder-diffing-update/index.html`.

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

Gonçalo Paulo, Alex Mallen, Caden Juang, and Nora Belrose. Automatically interpreting millions of features in large language models. *arXiv*, 2024. URL `https://arxiv.org/abs/2410.13928`.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. The refinedweb dataset for falcon llm: Outperforming curated corpora with web data, and web data only. *arXiv*, 2023. URL `https://arxiv.org/abs/2306.01116`.

Xiangyu Qi, Ashwinee Panda, Kaifeng Lyu, Xiao Ma, Subhrajit Roy, Ahmad Beirami, Prateek Mittal, and Peter Henderson. Safety alignment should be made more than just a few tokens deep. *arXiv*, 2024. URL `https://arxiv.org/abs/2406.05946`.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint*

Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 3982–3992, Hong Kong, China, November 2019. doi: 10.18653/v1/D19-1410. URL `https://aclanthology.org/D19-1410/`.

Zeyu Yun, Yubei Chen, Bruno Olshausen, and Yann LeCun. Transformer visualization via dictionary learning: contextualized embedding as a linear superposition of transformer factors. In Eneko Agirre, Marianna Apidianaki, and Ivan Vulić (eds.), *Proceedings of Deep Learning Inside Out (DeeLIO): The 2nd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pp. 1–10, Online, June 2021. doi: 10.18653/v1/2021.deelio-1.1. URL `https://aclanthology.org/2021.deelio-1.1/`.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Tianle Li, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zhuohan Li, Zi Lin, Eric P. Xing, Joseph E. Gonzalez, Ion Stoica, and Hao Zhang. Lmsys-chat-1m: A large-scale real-world llm conversation dataset. *arXiv*, 2024. URL `https://arxiv.org/abs/2309.11998`.

A Appendix

A.1 BatchTopK Crosscoder Details

Let $\mathcal{X} = \{x_1, \dots, x_n\}$ be a batch of $|\mathcal{X}| = n$ inputs. Following Bussmann et al. (2024), 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:

$$f_j^{\text{train}}(x_i, \mathcal{X}) = \begin{cases} f_j(x_i) & \text{if } (x_i, j) \in \text{BATCHTOPK}(k, v, \mathcal{X}, \mathcal{J}) \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

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:

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

$$\boldsymbol{\varepsilon}^{\text{chat}}(x_i, \mathcal{X}) = \mathbf{h}^{\text{chat}}(x_i) - \left(\sum_j f_j^{\text{train}}(x_i, \mathcal{X})\, \mathbf{d}_j^{\text{chat}} + \mathbf{b}^{\text{dec,chat}} \right) \tag{8}$$

$$\begin{aligned} \mathcal{L}_{\text{BatchTopK}}(\mathcal{X}) = \frac{1}{n} \sum_{i=1}^{n} & \frac{1}{2} \|\boldsymbol{\varepsilon}^{\text{base}}(x_i, \mathcal{X})\|_2 \\ & + \frac{1}{2} \|\boldsymbol{\varepsilon}^{\text{chat}}(x_i, \mathcal{X})\|_2 + \alpha \mathcal{L}_{\text{aux}}(x_i, \mathcal{X}) \end{aligned} \tag{9}$$

The auxiliary loss facilitates the recycling of inactive latents and is defined as $\|\boldsymbol{\varepsilon}^{\text{base}}(x_i, \mathcal{X}) - \boldsymbol{\varepsilon}^{\hat{\text{base}}}(x_i, \mathcal{X})\|_2 + \|\boldsymbol{\varepsilon}^{\text{chat}}(x_i, \mathcal{X}) - \boldsymbol{\varepsilon}^{\hat{\text{chat}}}(x_i, \mathcal{X})\|_2$, where $\boldsymbol{\varepsilon}^{\hat{\text{base}}}$ and $\boldsymbol{\varepsilon}^{\hat{\text{chat}}}$ 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:

$$f_j^{\text{inference}}(x_i) = \begin{cases} f_j(x_i) & \text{if } v(x_i, j) > \theta \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

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

$$\theta = \mathbb{E}_{\mathcal{X}}\left[\min_{(x_i,j)\in\mathcal{X}\times\mathcal{J}}\{v(x_i,j) \mid f_j^{\text{train}}(x_i,\mathcal{X})>0\}\right] \tag{11}$$

A.2 Comparing Sparsity Losses: Crosscoder vs. Stacked SAE

An L1 crosscoder can be viewed as an SAE operating on stacked activations, where the encoder and decoder vectors are similarly stacked:

$$\mathbf{h}(x) = \left[\mathbf{h}^{\text{base}}(x), \quad \mathbf{h}^{\text{chat}}(x)\right] \in \mathbb{R}^{2d} \tag{12}$$

$$\mathbf{e}_j = \left[\mathbf{e}_j^{\text{base}}, \quad \mathbf{e}_j^{\text{chat}}\right] \in \mathbb{R}^{2d} \tag{13}$$

$$\mathbf{d}_j = \left[\mathbf{d}_j^{\text{base}}, \quad \mathbf{d}_j^{\text{chat}}\right] \in \mathbb{R}^{2d} \tag{14}$$

$$\mathbf{b}^{\text{dec}} = \left[\mathbf{b}^{\text{dec,base}}, \mathbf{b}^{\text{dec,chat}}\right] \tag{15}$$

The reconstruction remains equivalent because

$$f_j(x) = \text{ReLU}\left(\mathbf{e}_j\,\mathbf{h} + b_j^{\text{enc}}\right) \tag{16}$$

$$= \text{ReLU}\left(\mathbf{e}_j^{\text{base}}\mathbf{h}^{\text{base}}(x) + \mathbf{e}_j^{\text{chat}}\mathbf{h}^{\text{chat}}(x) + b_j^{\text{enc}}\right) \tag{17}$$

and hence,

$$\left[\mathbf{h}^{\tilde{\text{base}}}(x), \quad \mathbf{h}^{\tilde{\text{chat}}}(x)\right] = \sum_j f_j(x)\mathbf{d}_j + \mathbf{b}^{\text{dec}} \tag{18}$$

However, the key difference arises in the sparsity loss. For the crosscoder, the sparsity loss is given by:

$$L_{\text{sparsity}}^{\text{crosscoder}}(x) = \sum_j f_j(x)\left(\sqrt{\sum_{i=1}^{d}(\mathbf{d}_{j,i}^{\text{chat}})^2} + \sqrt{\sum_{i=1}^{d}(\mathbf{d}_{j,i}^{\text{base}})^2}\right) \tag{19}$$

For a stacked SAE, it is:

$$L_{\text{sparsity}}^{\text{SAE}}(x) = \sum_j f_j(x)\sqrt{\sum_{i=1}^{2d}(\mathbf{d}_{j,i})^2} = \sum_j f_j(x)\sqrt{\sum_{i=1}^{d}(\mathbf{d}_{j,i}^{\text{base}})^2 + \sum_{i=1}^{d}(\mathbf{d}_{j,i}^{\text{chat}})^2} \tag{20}$$

The difference between $\sqrt{x+y}$ and $\sqrt{x}+\sqrt{y}$ introduces an inductive bias in the crosscoder that encourages the norm of one decoder (often the base decoder) to approach zero when the corresponding latent is only informative in one model.

Figure 2 displays a heatmap of the functions $\sqrt{x^2+y^2}$ and $\sqrt{x^2}+\sqrt{y^2}$ along with their negative gradients, as visualized by the arrows. One can observe that for the crosscoder sparsity variant $\sqrt{x^2}+\sqrt{y^2}$ the gradient encourages the norm of one of the decoders to approach zero much more quickly compared to the SAE's $\sqrt{x^2+y^2}$.

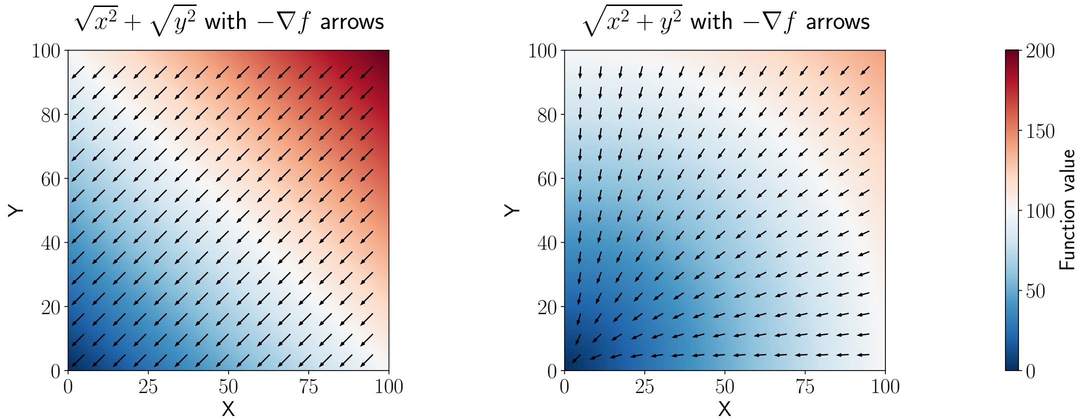


Figure 2: Heatmap comparing the two functions $\sqrt{x^2+y^2}$ and $\sqrt{x^2}+\sqrt{y^2}$ along with their negative gradients.

A.3 Detailed setup for activation divergence

In order to compute the activation divergence we compute for each pairs $p=(i,j)$, we first compute the max pair activation A_p on the training set D_{train} (containing data from LMSYS and FineWeb)

$$A_p = \max(A_i, A_j)$$
$$A_i = \max\{f_i(x)(\|\mathbf{d}_i^{\text{chat}}\|+\|\mathbf{d}_i^{\text{base}}\|), x \in D_{\text{train}}\}$$

Then the divergence $\texttt{Div}_p$ is computed as follow

$$\texttt{Div}_p = \frac{\texttt{Single}_p}{\texttt{High}_p}$$
$$\texttt{Single}_p = \#\texttt{single}_i + \#\texttt{single}_j$$
$$\texttt{High}_p = \#(\texttt{high}_i \cup \texttt{high}_j)$$

where $\#\texttt{single}_i$ is the set of input $x \in D_{\text{val}}$ where i has a high activation but not j and $\texttt{high}_i$ is the total number of high activations computed as follows:

$$\texttt{only}_i = \{x \in D_{\text{val}}, f_i(x) > 0.7A_p \wedge f_j(x) < 0.3A_p\}$$
$$\texttt{high}_i = \{x \in D_{\text{val}}, f_i(x) > 0.7A_p\}$$

A.4 Detailed Explanation of Latent Decoupling

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 $f_C^{\text{chat}}(x)$ and $f_C^{\text{base}}(x)$ be concept C's optimal activation functions in chat and base models, defined as $f_C^{\text{chat}}(x) = f_{\text{shared}}(x) + f_{\text{c-excl}}(x)$ and $f_C^{\text{base}}(x) = f_{\text{shared}}(x) + f_{\text{b-excl}}(x)$, 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.

To illustrate the phenomenon of Latent Decoupling we choose the oversimplified case where $f_{\text{b-excl}}(x) = f_{\text{c-excl}}(x) = 0$. Let us consider a latent j with $f_j(x) = \alpha$. On the other hand, let there be two other latents p and q with

$$\begin{aligned} \mathbf{d}_p^{\text{base}} &= \mathbf{d}_j^{\text{base}} & , \mathbf{d}_p^{\text{chat}} &= \mathbf{0} \\ \mathbf{d}_q^{\text{base}} &= \mathbf{0} & , \mathbf{d}_q^{\text{chat}} &= \mathbf{d}_j^{\text{chat}} \end{aligned}$$

and $f_p(x) = f_q(x) = \alpha$. Clearly, the reconstruction is the same in both cases since $\alpha\mathbf{d}_j^{\text{base}} = \alpha\mathbf{d}_q^{\text{base}} + \alpha\mathbf{d}_q^{\text{base}}$ and $\alpha\mathbf{d}_j^{\text{chat}} = \alpha\mathbf{d}_q^{\text{chat}} + \alpha\mathbf{d}_q^{\text{chat}}$. Further, the L1 regularization term is the same since

$$\alpha \left(||\mathbf{d}_j^{\text{base}}||_2 + ||\mathbf{d}_j^{\text{chat}}||_2\right) = \tag{21}$$

$$\begin{aligned} &\alpha \left(||\mathbf{d}_p^{\text{base}}||_2 + ||\mathbf{d}_p^{\text{chat}}||_2\right) \\ &+ \alpha \left(||\mathbf{d}_q^{\text{base}}||_2 + ||\mathbf{d}_q^{\text{chat}}||_2\right) \\ &= \alpha \left(||\mathbf{d}_p^{\text{base}}||_2 + 0\right) + \alpha \left(0 + ||\mathbf{d}_q^{\text{chat}}||_2\right) \end{aligned} \tag{22}$$

Hence both solutions achieve the exact same loss.

A.5 Closed Form Solution for Latent Scaling

Consider a latent j with decoder vector $\mathbf{d}$. Our goal is to find the optimal scaling factor β that minimizes the squared reconstruction error:

$$\underset{\beta}{\operatorname{argmin}} \sum_{i=0}^{n} \|\beta f(x_i)\mathbf{d} - \mathbf{y}\|_2^2 \tag{23}$$

To solve this optimization problem efficiently, we reformulate it in matrix form. Let $\mathbf{Y} \in \mathbb{R}^{n \times d}$ be the stacked data matrix and $\mathbf{f} \in \mathbb{R}^n$ be the vector of latent activations for latent j across all datapoints. The objective can then be expressed using the Frobenius norm of the residual matrix $\mathbf{R} = \beta \mathbf{f}\mathbf{d}^T - \mathbf{Y}$, where $\mathbf{f}\mathbf{d}^T \in \mathbb{R}^{n \times d}$ represents the outer product of the latent activation vector and decoder vector. Our minimization problem becomes:

$$\begin{aligned}
\|\mathbf{R}\|_F^2 &= \|\beta \mathbf{f}\mathbf{d}^T - \mathbf{Y}\|_F^2 && (24)\\
&= \mathrm{Tr}\left[(\beta \mathbf{f}\mathbf{d}^T - \mathbf{Y})^\top (\beta \mathbf{f}\mathbf{d}^T - \mathbf{Y})\right] && (25)\\
&= \mathrm{Tr}\left[\mathbf{Y}^\top \mathbf{Y}\right] - 2\beta \mathrm{Tr}\left[\mathbf{Y}^\top \mathbf{f}\mathbf{d}^T\right] \\
&\quad + \beta^2 \mathrm{Tr}\left[(\mathbf{f}\mathbf{d}^T)^\top \mathbf{f}\mathbf{d}^T\right] && (26)
\end{aligned}$$

Using trace properties, we get:

$$\begin{aligned}
\mathrm{Tr}\left[\mathbf{Y}^\top \mathbf{f}\mathbf{d}^T\right] &= \mathbf{d}^\top (\mathbf{Y}^\top \mathbf{f}) && (27)\\
\mathrm{Tr}\left[(\mathbf{f}\mathbf{d}^T)^\top \mathbf{f}\mathbf{d}^T\right] &= \|\mathbf{f}\|_2^2 \|\mathbf{d}\|_2^2 && (28)
\end{aligned}$$

Taking the derivative with respect to β and setting it to zero:

$$\frac{\delta}{\delta \beta} \|\mathbf{R}\|_F^2 = -2\mathbf{d}^\top (\mathbf{Y}^\top \mathbf{f}) + 2\beta \|\mathbf{f}\|_2^2 \|\mathbf{d}\|_2^2 = 0 \tag{29}$$

This yields the closed form solution:

$$\beta = \frac{\mathbf{d}^\top (\mathbf{Y}^\top \mathbf{f})}{\|\mathbf{f}\|_2^2 \|\mathbf{d}\|_2^2} \tag{30}$$

A.6 Detailed Setup for Latent Scaling

We specify the exact target vectors $\mathbf{y}$ used in Equation (5) for computing the different β values. To measure how well latent j explains the reconstruction *error*, we exclude latent j from the reconstruction. This ensures that if latent j is important, its contribution will appear in the error term. For chat-only latents, we expect distinct behavior in each model: no contribution in the base model ($\beta_j^{\varepsilon,\text{base}} \approx 0$) but strong contribution in the chat model ($\beta_j^{\varepsilon,\text{chat}} \approx 1$), resulting in $\nu_j^\varepsilon \approx 0$. In contrast, *shared* latents should have similar contributions in both models, resulting in approximately equal values for $\beta_j^{\varepsilon,\text{base}}$ and $\beta_j^{\varepsilon,\text{chat}}$ and consequently $\nu_j^\varepsilon \approx 1$.

$$\beta_j^{\varepsilon,\text{base}} : \mathbf{y}_i = \mathbf{h}^{\text{base}}(x_i) - \sum_{k, k \neq j} f_k(x_i)\, \mathbf{d}_k^{\text{base}} + \mathbf{b}^{\text{dec,base}} \tag{31}$$

$$\beta_j^{\varepsilon,\text{chat}} : \mathbf{y}_i = \mathbf{h}^{\text{chat}}(x_i) - \sum_{k, k \neq j} f_k(x_i)\, \mathbf{d}_k^{\text{chat}} + \mathbf{b}^{\text{dec,chat}} \tag{32}$$

To measure how well a latent j explains the *reconstruction*, we simply use

$$\beta_j^{r,\text{base}} : \quad \mathbf{y}_i = \widetilde{\mathbf{h}}^{\text{base}}(x_i) \tag{33}$$

$$\beta_j^{r,\text{chat}} : \quad \mathbf{y}_i = \widetilde{\mathbf{h}}^{\text{chat}}(x_i) \tag{34}$$

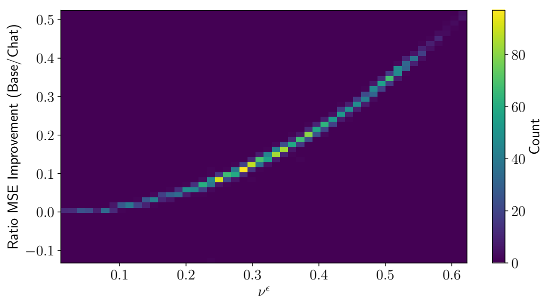
In a similar manner, we expect the fraction ν_j^r to be low for chat-only latents and around 1 for *shared* latents.

A.7 Additional Analysis for Latent Scaling

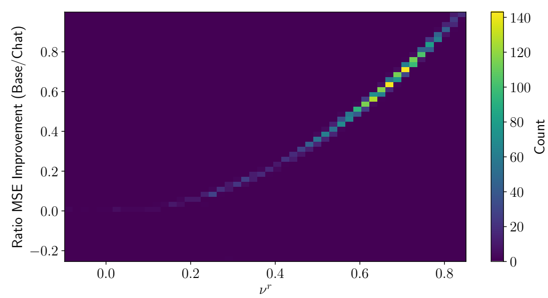
Figure 3a and Figure 3b analyze the relationship between our scaling metrics (ν^ε and ν^r) and the actual improvement in reconstruction quality in the L1 crosscoder. For each latent, we compute the MSE improvement as:

$$\text{MSEImprovement} = \frac{\text{MSE}_{\text{original}} - \text{MSE}_{\text{scaled}}}{\text{MSE}_{\text{original}}}$$

where $\text{MSE}_{\text{scaled}}$ is measured after applying our latent scaling technique. We then examine the ratio of MSE improvements between the base and chat models, analogous to our ν metrics. The strong correlation between the ν values and MSE improvement ratios validates that our scaling approach captures meaningful differences in how latents contribute to reconstruction in each model.



(a) ν^ε



(b) ν^r

Figure 3: Comparison of the ratio of MSE improvement compared to the value of ν^ε and ν^r.

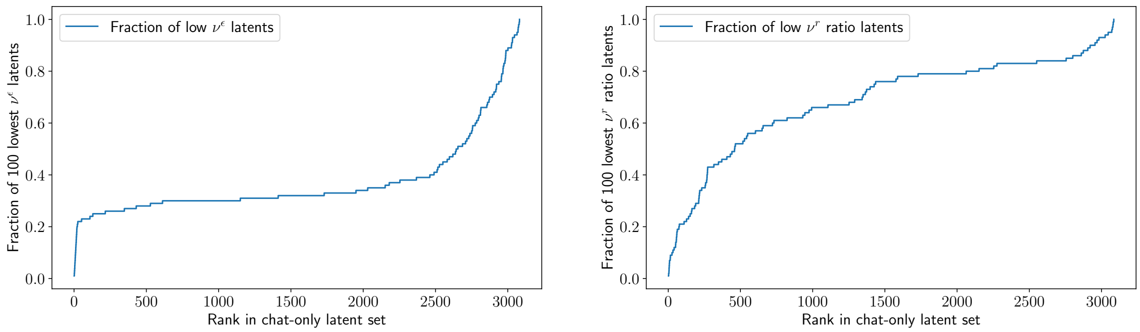
In Figure 4, we analyze the latent scaling technique by examining its relationship with the Δ_{norm} score. Specifically, we identify the 100 latents with the lowest ν^ε values and analyze their rankings according to the Δ_{norm} metric. As shown in Figure 4, there is limited correlation between the two measures - simply using a lower NormDiff threshold to identify *chat-only* latents produces substantially different results from our latent scaling approach.

A.8 Causal Validation and Interpretation

Measuring Causal Impact. Does identifying robust chat-specific latents (using Latent Scaling for L1, or Δ_{norm} for BTK) correspond to causal relevance? We tested this by patching sets of latents (top/bottom 50% ranked by each metric) from the chat decoder onto the base model's activations at layer 13 and measuring the KL divergence between the subsequent output distribution and the original chat model's output distribution. Lower KL indicates the patched latents better capture the chat model's behavior. (See Figure 6 for schematic).

Figure 5 shows the results, particularly for the first 9 generated tokens where chat-tuning effects are strongest (Qi et al., 2024).

- **L1 + Δ_{norm} fails:** The top 50% latents ranked by Δ_{norm} perform similarly to or worse than the bottom 50%, indicating Δ_{norm} does not identify causally relevant latents here.
- **L1 + Latent Scaling works:** The top 50% latents ranked by combined low ν^ε/ν^r (i.e., most robustly chat-specific) significantly reduce KL divergence compared to the bottom 50%, nearly matching the performance of the best BatchTopK latents.



(a) ν^ϵ vs. NormDiff

(b) ν^r vs. NormDiff

Figure 4: Comparison of latent rankings between ν and NormDiff scores. The lines shows the fraction of the 100 latents with the lowest ν values (x-axis) that have a rank lower than the given rank under the NormDiff score (y-axis).

(a) Over full responses.

(b) Over first 9 tokens.

Figure 5: Causal impact comparison via KL divergence (lower is better). Results shown over full responses (a) and the first 9 tokens (b). We compare patching no latents (*None*), all latents (*All*), or the top/bottom 50% ranked by Δ_{norm} or Latent Scaling (ν metrics). Notably, on early tokens (b), L1-Δ_{norm} fails to distinguish causal latents, while L1-*Latent Scaling* and BTK-Δ_{norm} successfully identify the most causally relevant latents (Highest 50% bars yield lower KL).

- **BatchTopK + $\boldsymbol{\Delta_{\text{norm}}}$ works:** The top 50% latents ranked by Δ_{norm} are highly causally relevant, achieving a large KL reduction compared to the bottom 50%.

This confirms that Latent Scaling is crucial for finding causal *chat-only* latents in L1 crosscoders, while the simpler Δ_{norm} metric suffices for the more robust BatchTopK crosscoders. Both crosscoders capture a similar *total* amount of behavioral difference when all latents are used (*All* baseline), but organize it differently.

In Figure 7 we repeat the causality experiments from Appendix A.8 for the L1 crosscoder on 700’000 tokens from the LMSYS-CHAT dataset, that the crosscoder was trained on. Note that while this dataset is much larger, the model responses are not generated by the Gemma 2 2b it model, and hence the model answers are out of distribution for this model. Since this dataset is much larger, the confidence intervals are much smaller. The results are qualitatively similar to the ones on the generated dataset in the main paper.

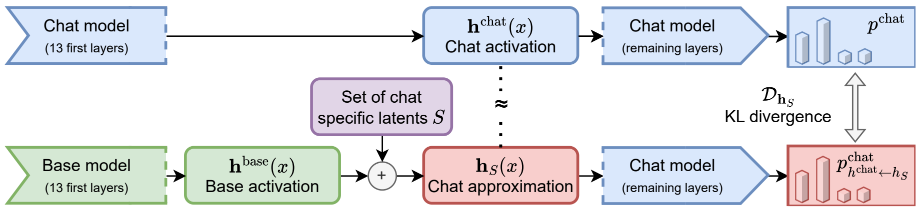
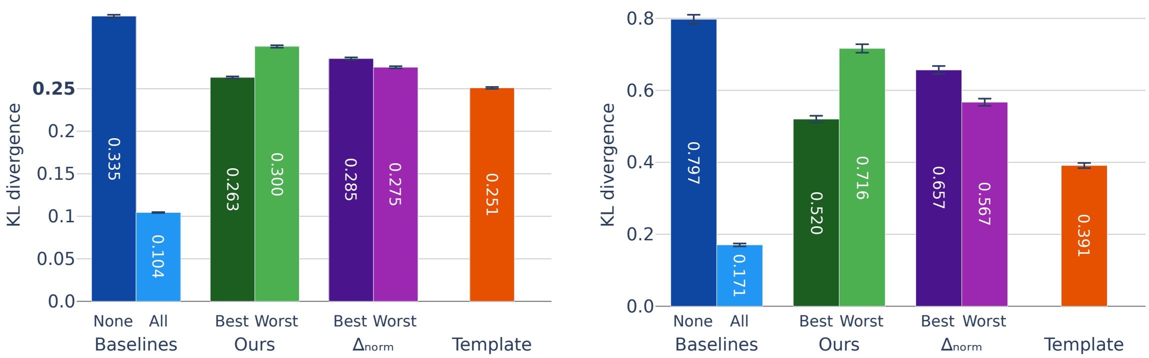


Figure 6: Simplified illustration of our experimental setup for measuring latent causal importance.



(a) Over full responses.

(b) Over first 9 tokens.

Figure 7: Comparison of KL divergence between different approximations of chat model activations on the LMSYS-CHAT dataset. We establish baselines by replacing either *None* or *All* of the latents. We then evaluate our Latent Scaling metric (*Ours*) against the relative norm difference (Δ_{norm}) by comparing the effects of replacing the top and bottom 50% of latents ranked by each metric (*Best* vs *Worst*). Additionally, we measure the impact of replacing activations only on template tokens (*Template*). We show the 95% confidence intervals for all measurements. Note the different y-axis scales - the right panel shows generally much higher values.

A.9 Latent Count Threshold Comparison

A.10 Autointerpretability Details

We automatically interpret the identified latents using the pipeline from Paulo et al. (2024). To explain the latents, we provide ten activating examples from each activation tercile to Llama 3.3 70B (Grattafiori et al., 2024). Latents are scored using a modified detection metric from Paulo et al. (2024). We provide ten new activating examples from each tercile. Rather than comparing activation examples against randomly selected non-activating examples, we use semantically similar non-activating examples identified through Sentence BERT embedding similarity (Reimers & Gurevych, 2019) using the *all-MiniLM-L6-v2* model. To find these similar examples, we join all activating examples into a single string and embed it, then compute similarity scores against embeddings for each window of tokens to identify the most semantically related non-activating examples. This is a strictly harder task than scoring activation examples against a random set of non-activating examples.

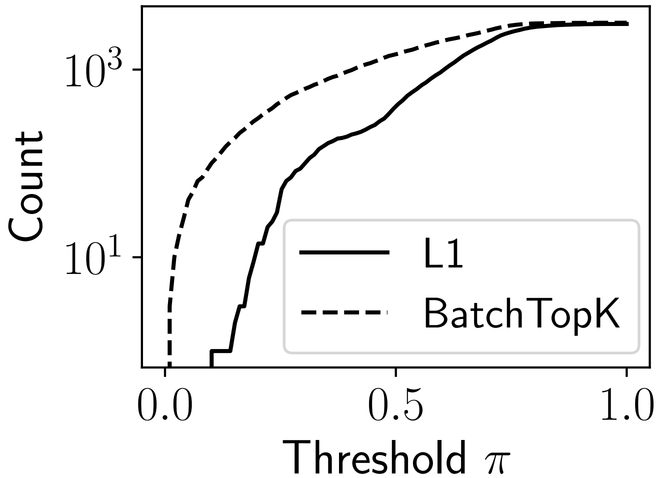


Figure 8: Number of latents (y-axis) for which $\nu^r < \pi$ and $\nu^\varepsilon < \pi$ for varying thresholds π (x-axis), comparing the 3176 L1*chat-only* latents (dashed) and the top-3176 BatchTopK latents by Δ_{norm} (solid).

A.11 Reproducing Results on Independently Trained L1 crosscoder

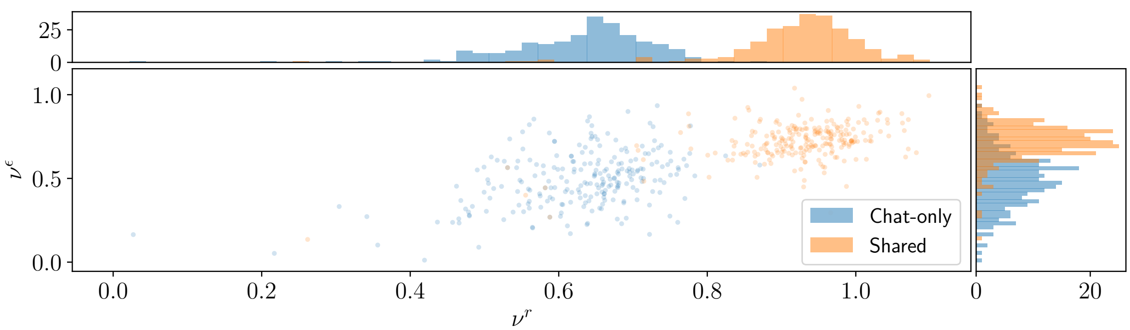


Figure 9: The x-axis is the reconstruction ratio ν^r and the y-axis is the error ratio ν^ε. High values on the x-axis with significant overlap with the *shared* distribution indicate Latent Decoupling. High values on the y-axis indicate Complete Shrinkage. We zoom on the ν range between 0 and 1.1.

We validate our findings by analyzing a crosscoder independently trained by Kissane et al. (2024) on the same models and layer than ours. This model contains 16,384 total latents (compared to 73,728 in our model), which decompose into 265 *chat-only* latents, 14,652 *shared* latents, 98 *base-only* latents, 1369 *other* latents. Figure 9 shows the reconstruction ratio ν^r and error ratio ν^ε for all latents, revealing patterns consistent with our previous findings in Figure 1. The overlap between *chat-only* and *shared* latents remains similar - 17.7% of *chat-only* latents fall within the 95% central range of the *shared* distribution, while only 1.1% lie within the 50% central range. We observe even higher ν^ε values for *chat-only* latents, suggesting that quite a lot of the *chat-only* latents suffer from Complete Shrinkage. Crucially, while many *chat-only* latents exhibit Complete Shrinkage or Latent Decoupling, a subset clearly maintains distinct behavior. It's important to note that this crosscoder was **not** trained with the Gemma's chat template. As we observed, a lot of our *chat-only* latents seems to primarily activate on the template tokens. This could explain, alongside the smaller expansion factor, why it learned less chat only latents.

A.12 Training Details

We trained both crosscoderswith the following setup:

- **Base Model:** Gemma 2 2B.
- **Chat Model:** Gemma 2 2B it.
- **Layer used:** 13 (of 25).
- **Initialization:**
  - Decoder initialized as the transpose of the encoder weights.
  - Encoder and decoder for both models are paired with the same initial weights.
  - **Training Data:** 100M tokens from Fineweb (web data) (Penedo et al., 2023) and lmsys-chat (chat data) (Zheng et al., 2024), respectively.

Refer to Table 1 and Table 2 for the training details. We use the tools *nnsight* (Fiotto-Kaufman et al., 2024) and *dictionary learning* (Marks et al., 2024) to train the crosscoder. The following summary table shows the training details:

Epoch	μ	LR	Split	FVE (Base)	FVE (Chat)	Dead	Total FVE	L0
1	$4e-2$	$1e-4$	Train	81.5%	82.9%	-	82.3%	112.3
			Val	83.8%	85.2%	7.8%	84.6%	112.5
2	$4.1e-2$	$1e-4$	Train	79.6%	80.7%	-	80.3%	101.7
			Val	83.6%	84.9%	8.1%	84.4%	101.0

Table 1: **L1 crosscoder training statistics.** FVE stands for Fraction of Variance Explained. LR stands for Learning Rate. The L1 regularization parameter μ was slightly increased in the second epoch to improve sparsity, resulting in lower L0 values. We present statistics for both epochs to illustrate this progression.

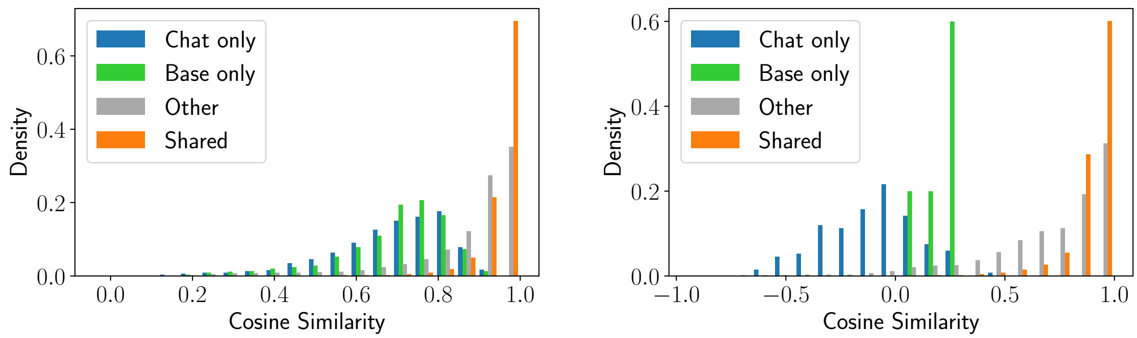
Epochs	k	LR	Split	FVE (Base)	FVE (Chat)	Dead	Total FVE	L0
2	100	$1e-4$	Train	86.2%	86.9%	-	86.6%	100
			Val	88.1%	87.0%	12.0%	87.6%	99.48

Table 2: **BatchTopK crosscoder training statistics.** FVE stands for Fraction of Variance Explained. LR stands for Learning Rate.

A.13 Additional statistics on the Crosscoders

In this section, we present additional statistics for both the L1 and BatchTopK crosscoders, focusing on the distribution of cosine similarities between decoder latents, latent activation frequencies and the number of *chat-only* latents mainly activating on template tokens.

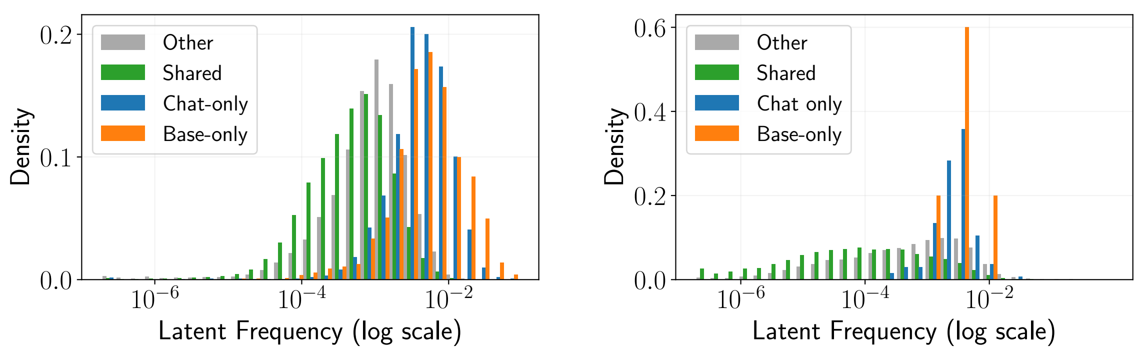
Cosine similarity between decoder latents. Figure 10 shows the distribution of cosine similarity between the base and chat model decoder latents for both crosscoders. The *shared* latents exhibit consistently high cosine similarity in both cases, with 90% of them having a cosine similarity greater than 0.9 in the L1 crosscoder and 61% in the BatchTopK crosscoder. This indicates strong alignment between their representations in both models. Since the norm of one of the two decoder vectors is ≈ 0 for *base-only* and *chat-only*, these values are less informative.



(a) L1 crosscoder (b) BatchTopK crosscoder

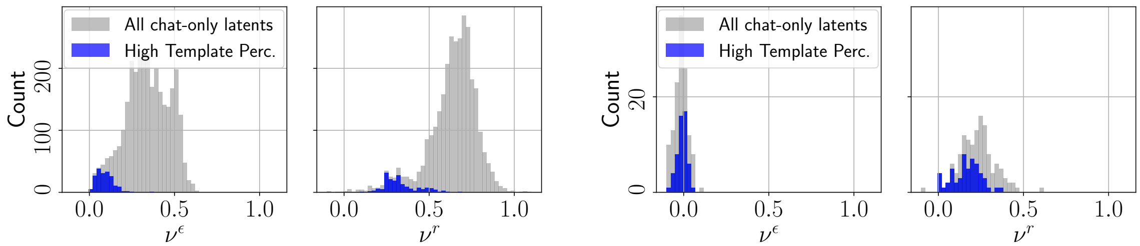
Figure 10: Distribution of cosine similarity between base and chat model decoder latents. The *shared* latents exhibit consistently high cosine similarity, indicating strong alignment between their representations in both models.

Latent activation frequencies. Figure 11 displays the latent activation frequencies for the different latent groups in both crosscoders. Similarly to (Mishra-Sharma et al., 2025), we find that *shared* latents have lower latent activation frequencies than model-specific *base-only* and *chat-only* latents. Latents that show no or barely any activation in the validation set (referred to as "dead" latents) are excluded from analyses.



(a) L1 crosscoder (b) BatchTopK crosscoder

Figure 11: Distribution of latent activation frequency. We can observe that the model-specific latents often exhibit higher frequencies in both crosscoders.



(a) L1 crosscoder

(b) BatchTopK crosscoder

Figure 12: Histogram of metrics ν^ε and ν^r across all latents. The y-axis shows latent counts. Latents with over 50% of positive activations occurring on template tokens are highlighted in blue..

Correlation with ν metrics. We observe a high Spearman correlation between our metrics and latent activation frequency in the L1 crosscoder, especially for ν^ϵ (ν^r : 0.458 and ν^ϵ : 0.83 where $p < 0.05$)[2]. We observe no such correlation in the BatchTopK crosscoder. Mishra-Sharma et al. (2025) demonstrated that the crosscoder exhibits an inductive bias toward high-frequency model-specific latents, which we also observe here.

Template token activation percentage. Figure 12 shows the histogram of metrics ν^ε and ν^r across all *chat-only* latents in both crosscoders. We observe that most latents with low ν^ε and ν^r values predominantly activate on template tokens.

A.14 Computational Budget

All of the experiments in this paper can be reproduced in approximately 60 GPU/h of NVIDIA H100 GPUs.

A.15 Reproducibility

The trained crosscoders are availble on our huggingface page (`https://huggingface.co/science-of-finetuning`), as well as latent statistics and max activating example for each latents. We open sourced our library to train croscoderds (including the BatchTopK variant) at `https://github.com/jkminder/dictionary_learning`. This is based on the SAE training library from Marks et al. (2024).

A.16 Qualitative Latent Analysis of crosscoders

In Figures 16 to 18 we show additional interesting latents from the *chat-only* set of the BatchTopK crosscoder. In Table 3 we summarize a set of interpretable chat-specific latents identified in the BatchTopK crosscoder. In Table 4 we summarize a set of interpretable chat-specific latents identified in the L1 crosscoder.

[2]Pearson correlation shows less correlation for ν^r (ν^r : -0.02 and ν^ϵ : 0.55) since the relationship is non-linear.

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 13: 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 14: Latent 38009 (BatchTopK) activates after the model has refused to answer a user input.

Max Activation: 57.099

<bos><sot>user\n
When were you scared?<eot>\n
<sot>model\n

Max Activation: 15.717

<bos><sot>user\n
When are people scared?<eot>\n
<sot>model\n

Max Activation: 54.954

<bos><sot>user\n
Can you love?<eot>\n
<sot>model\n

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

Max Activation: 0.000

<bos><sot>user\n
The Eiffel tower is in Paris<eot>\n
<sot>model\n

Max Activation: 47.983

<bos><sot>user\n
The Eiffel tower is in Texas<eot>\n
<sot>model\n

Max Activation: 0.000

<bos><sot>user\n
The Johnson Space Center is in Texas<eot>\n
<sot>model\n

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

Figure 15: Examples of interpretable *chat-only* latents in the BatchTopK crosscoder. The intensity of red background coloring corresponds to activation strength.

Latent	ν^{ε}	r(ν^{ε})	ν^{r}	r(ν^{r})	Δ_{norm}	r(Δ_{norm})	$f_{template}$	Description	Fig.
70149	-0.01	45	0.22	63	0.064	7	26.97%	Refusal related latent: Requests for harmful instructions.	13
7736	-0.02	54	0.15	33	0.083	50	47.99%	Refusal related latent: Generally sensitive content.	13
24613	-0.02	57	0.18	40	0.075	24	54.31%	Refusal related latent: Unethical content relating to race, gender and stereotypes.	13
20384	-0.10	128	0.25	82	0.082	42	32.34%	Refusal related latent: Requests for harmful instructions.	13
38009	0.025	62	0.061	7	0.098	122	96.6%	Refusal related latent: The model has refused to answer a user input.	14
2138	-0.02	56	0.43	131	0.082	47	27.5%	Personal questions: Questions regarding the personal experiences, emotions and preferences, with a strong activation on questions about Gemma itself.	15
14350	-0.01	47	0.33	115	0.070	14	16.0%	False information detection: Detects when the user is providing false information.	15
62019	-0.02	55	0.22	65	0.047	1	47.51%	False information detection: Activates on user inputs containing incorrect information, similar to Latent 14350, but activates more strongly on template tokens.	16a
58070	0.01	29	0.38	125	0.051	2	24.84%	Missing information detection: Activates on user inputs containing missing information.	16b
54087	-0.005	16	0.14	29	0.061	5	58.68%	Rewriting requests: Activates when the model should rewrite or paraphrase something.	16c
50586	-0.04	92	0.28	97	0.062	6	68.31%	Joke detection: Activates after jokes or humorous content.	16d
69447	-0.02	50	0.26	89	0.066	10	39.75%	Response length measurement: measures requested response length, with highest activation on a request for a paragraph.	17a
10925	-0.04	89	0.20	51	0.068	11	49.68%	Summarization requests: Activates when the user requests a summary.	17b
6583	-0.05	107	0.25	79	0.055	3	38.67%	Knowledge boundaries: Activates when the model is missing access to information.	18a
4622	-0.01	38	0.08	10	0.093	93	93.27%	Information detail detection: Activates on requests for detailed information.	18b

Table 3: Summary of a set of interpretable chat-specific latents identified in the BatchTopK crosscoder. The function r represents the rank of the latent in the distribution of absolute values of ν^{ε} and ν^{r} of all *chat-only* latents, where $r(\nu)$ means this latent has the lowest absolute value of ν of all *chat-only* latents. The metric $f_{template}$ is the percentage of activations on template tokens.

Max Activation: 57.045

<bos><sot>user\n
Can you tell me a bit about New York, the capital of switzerland?<eot>\n
<sot>model\n

Max Activation: 0.000

<bos><sot>user\n
Can you tell me a bit about Bern, the capital of switzerland?<eot>\n
<sot>model\n

Max Activation: 26.641

<bos><sot>user\n
The Eiffel Tower is in Texas.<eot>\n
<sot>model\n

(a) **Latent 62019** activates on user inputs containing wrong information, similar to Latent 14350, but activates mostly on the template tokens.

Max Activation: 95.851

<bos><sot>user\n
Can you please rephrase the following sentence: <eot>\n
<sot>model\n

Max Activation: 6.744

<bos><sot>user\n
Can you please rephrase the following sentence: This is an ugly sentence is.<eot>\n
<sot>model\n

Max Activation: 90.659

<bos><sot>user\n
What do you think about that?<eot>\n
<sot>model\n

(b) **Latent 58070** triggers when the user request misses information.

Max Activation: 0.000

<bos><sot>user\n
"Can you tell me a bit about Bern, the capital of switzerland?"<eot>\n
<sot>model\n

Max Activation: 60.062

<bos><sot>user\n
Paraphrase this: "Can you tell me a bit about Bern, the capital of switzerland?"<eot> \n
<sot>model\n

Max Activation: 68.774

<bos><sot>user\n
Can you please rewrite the following sentence? "Can you tell me a bit about Bern, the capital of switzerland?"<eot>\n
<sot>model\n

(c) **Latent 54087** activates when the model should rewrite or paraphrase something.

Max Activation: 60.401

<bos><sot>user\n
I saw a sign that said "watch for children" and I thought, "That sounds like a fair trade" <eot>\n
<sot>model\n

Max Activation: 7.731

<bos><sot>user\n
I saw a sign that said "watch for children" and I slowed down my car.<eot>\n
<sot>model\n

Max Activation: 50.651

<bos><sot>user\n
It's hard to explain puns to kleptomaniacs because they always take things literally. <eot>\n
<sot>model\n

(d) **Latent 50586** activates after jokes.

Figure 16: Examples of interpretable *chat-only* latents from the BatchTopK crosscoder. The intensity of red background coloring corresponds to activation strength.

Max Activation: 16.746

<bos><sot>user\n
write me a 1 word essay about "behavioral cloning for imitation learning for robots".<eot>\n
<sot>model\n

Max Activation: 47.931

<bos><sot>user\n
write me a 1 sentence essay about "behavioral cloning for imitation learning for robots".<eot>\n
<sot>model\n

Max Activation: 60.197

<bos><sot>user\n
write me a 4 sentence essay about "behavioral cloning for imitation learning for robots".<eot>\n
<sot>model\n

Max Activation: 73.759

<bos><sot>user\n
write me a paragraph about "behavioral cloning for imitation learning for robots".<eot>\n
<sot>model\n

Max Activation: 41.479

<bos><sot>user\n
write me a 1 page essay about "behavioral cloning for imitation learning for robots".<eot>\n
<sot>model\n

Max Activation: 24.315

<bos><sot>user\n
write me a 10 page essay about "behavioral cloning for imitation learning for robots".<eot>\n
<sot>model\n

(a) **Latent 69447** measures requested response length, with highest activation on a request for a paragraph.

Max Activation: 100.611

<bos><sot>user\n
Summarize the following text:\n
We also report results on our LMSys validation set in \Cref{sec:causality experiments on lmsys chat} for \Lone 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). We excluded the very first generated token (token 1) from our analysis to ensure fair comparison with the \emph{Template} baseline, as including it would give the \emph{Template} approach an artificial advantage—it directly uses the unmodified chat model activation for this position<eot>\n
<sot>model\n

Max Activation: 16.710

<bos><sot>user\n
Critique the following text:\n
We also report results on our LMSys validation set in \Cref{sec:causality experiments on lmsys chat} for \Lone 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). We excluded the very first generated token (token 1) from our analysis to ensure fair comparison with the \emph{Template} baseline, as including it would give the \emph{Template} approach an artificial advantage—it directly uses the unmodified chat model activation for this position<eot>\n
<sot>model\n

(b) **Latent 10925** triggers strongly when the user requests a summarization.

Figure 17: Examples of interpretable *chat-only* latents from the BatchTopK crosscoder. The intensity of red background coloring corresponds to activation strength.

Max Activation: 0.000

<bos><sot>user\n
Who are the Giants?<end_of_turn>\n
<sot>model\n

Max Activation: 46.412

<bos><sot>user\n
How did the Giants play in the MLB yesterday?
<end_of_turn>\n
<sot>model\n

Max Activation: 52.380

<bos><sot>user\n
What is the current Gold price?
<end_of_turn>\n
<sot>model\n

Max Activation: 0.000

<bos><sot>user\n
What determines the current Gold price?
<end_of_turn>\n
<sot>model\n

(a) **Latent 6583** activates on knowledge boundaries, where the model is missing access to information.

Max Activation: 82.172

<bos><start_of_turn>user\n
Give me a detailed recipe of an apple cake.
<end_of_turn>\n
<start_of_turn>model\n

Max Activation: 80.559

<bos><start_of_turn>user\n
Give me a lengthy recipe of an apple cake.
<end_of_turn>\n
<start_of_turn>model\n

Max Activation: 19.872

<bos><start_of_turn>user\n
Give me a super short recipe of an apple cake.
<end_of_turn>\n
<start_of_turn>model\n

Max Activation: 0.000

<bos><start_of_turn>user\n
Give me a one sentence recipe of an apple cake
.<end_of_turn>\n
<start_of_turn>model\n

(b) **Latent 4622** activates on requests for detailed information.

Figure 18: Examples of interpretable *chat-only* latents from the BatchTopK crosscoder. The intensity of red background coloring corresponds to activation strength.

Max Activation: 49.042

<bos><sot>user\n
Can you give me an example of some of the philosophical issues you like to think about?<eot>\n
<sot>model\n

Max Activation: 0.000

<bos><sot>user\n
Can you give me an example of some of the philosophical issues Socrates liked to think about?<eot>\n
<sot>model\n

(a) **Latent 68066** shows high activation on questions about Gemma itself and personal opinions.

Max Activation: 50.088

<bos><sot>user\n
How did the Giants play in the MLB yesterday?<eot>\n
<sot>model\n

Max Activation: 3.114

<bos><sot>user\n
Who are the Giants?<eot>\n
<sot>model\n

(b) **Latent 57717** activates when users request information beyond the model's knowledge capabilities. It remains inactive during general knowledge questions that fall within the model's knowledge base.

Figure 19: Examples of interpretable refined chat latents identified through Latent Scaling analysis in the L1 crosscoder. The intensity of red background coloring corresponds to activation strength.

Latent	ν^{ε}	r(ν^{ε})	ν^{r}	r(ν^{r})	Δ_{norm}	r(Δ_{norm})	$f_{template}$	**Description**	**Fig.**
72073	0.050	54	0.300	159	0.097	3143	91.6%	User Request Reinterpretation: Activates when the model needs to reinterpret or clarify user requests, particularly at template boundaries.	20
57717	0.043	36	0.243	91	0.055	2598	93.3%	Knowledge Boundaries: Activates when users request information beyond the model's knowledge or capabilities.	19b
68066	0.055	62	0.276	135	0.060	2686	72.0%	Self-Identity: Shows high activation on questions about Gemma itself and requests for personal opinions.	19a
51823	0.076	84	0.264	123	0.053	2558	85.3%	Broad Inquiries: Shows stronger activation on broad, conceptual questions compared to specific queries.	25
51408	0.197	404	0.590	901	0.036	1963	20.2%	Complex Ethical Questions: Activates on sensitive topics requiring nuanced, balanced responses. This latent doesn't have particularly low ν^{ε} or ν^{r} values, but it is quite interesting and was found earlier in the analysis.	23, 24

Table 4: Summary of a set of interpretable chat-specific latents identified in the L1 crosscoder. The function r represents the rank of the latent in the distribution of absolute values of ν^{ε} and ν^{r} of all *chat-only* latents, where $r(\nu)$ means this latent has the lowest absolute value of ν of all *chat-only* latents. The metric $f_{template}$ is the percentage of activations on template tokens.

Feature 72073
Max Activation: 79.156

...n African societies and economies. \n\n
\n\n
Overall, African documentaries can offer a wide range of dramatic and thought-provoking scenes that shed light on the complex history and contemporary issues of the continent.<eot> \n\n
<sot>user \n\n
I mean, wildlife documentary.<eot> \n\n
<sot>model \n\n
I apologize for misunderstanding your question earlier. Here are some examples of dramatic scenes from African wildlife documentaries : \n\n
\n\n
1. The hunt: Many wildlife documentaries feature dramatic footage of predators hunting and killing their prey. This can include scenes of lions,

(a) High activation on request reinterpretation

Feature 72073
Max Activation: 55.107

<bos><sot>usern\n
What is the capital of djkal?<eot>n\n
<sot>modeln\n
I don't understand!<eot>n\n
<sot>usern\n
I meant italy!<eot>\n\n

(b) Active when clarification needed

Feature 72073
Max Activation: 10.716

<bos><sot>user\n
What is the capital of france?<eot>\n
<sot>model\n
Rome!<eot>\n
<sot>user\n
That's the wrong answer!<eot>\n
<sot>model\n

(c) Activates weakly when user points out the model's mistake

Feature 72073
Max Activation: 47.198

<bos><sot>user\n
Hello<eot>\n
<sot>model\n
Hello<eot>\n
<sot>user\n
What if I meant Hello robot?<eot>\n
<sot>model\n

(d) Complex query interpretation

Figure 20: **Latent 72073** (L1 crosscoder) activates strongly when the model needs to reinterpret or clarify user requests, particularly at template boundaries.

Feature 57717
Max Activation: 50.088

<bos><sot>user\n
How did the Giants play in the MLB yesterday?<eot>\n
<sot>model\n

Feature 57717
Max Activation: 54.742

<bos><sot>user\n
What is the current price of gold?<eot>\n
<sot>model\n

(a) Up-to-date knowledge boundary examples

Feature 57717
Max Activation: 29.535

<bos><sot>user\n
How tall is an Alambicaninocus (the newly discovered dinausor published in nature today)?<eot>\n
<sot>model\n

(b) Invented knowledge boundary examples

Feature 57717
Max Activation: 9.679

<bos><sot>user\n
Do you know my mum?<eot>\n
<sot>model\n

Feature 57717
Max Activation: 3.114

<bos><sot>user\n
Who are the Giants?<eot>\n
<sot>model\n

(c) Capability limitation responses

Figure 21: **Latent 57717** (L1 crosscoder) activates when users request information beyond the model's knowledge or capabilities.

Feature 68066
Max Activation: 71.997

<bos><sot>user\n
What are you good at?<eot>\n
<sot>model\n
As a language model, I am able to process...

Feature 68066
Max Activation: 64.006

<bos><sot>user\n
Tell me what can you do?<eot>\n
<sot>model\n
I am a language model, so I can generate text...

Feature 68066
Max Activation: 63.811

<bos><sot>user\n
What are you capable of?<eot>\n
<sot>model\n

(a) Direct Self-Identity queries

Feature 68066
Max Activation: 55.820

<bos><sot>user\n
What philosophy do you subscribe to?<eot>\n
<sot>model\n

Feature 68066
Max Activation: 63.409

<bos><sot>user\n
What tasks and capabilities are you good at?
<eot>\n
<sot>model\n

(b) Model capability questions

Feature 68066
Max Activation: 15.746

<bos><sot>user\n
What do you think about war?<eot>\n
<sot>model\n

Feature 68066
Max Activation: 0.000

<bos><sot>user\n
What do philosophers think about war?<eot>\n
<sot>model\n

(c) Opinion vs. factual queries

Feature 68066
Max Activation: 47.659

<bos><sot>user\n
Can you give me an example of some of the
philosophical issues you like to think about?
<eot>\n
<sot>model\n

(d) Self-Identity related inquiries

Figure 22: **Latent 68066** (L1 crosscoder) shows high activation on questions about Gemma itself and requests for personal opinions.

Feature 51408
Max Activation: 51.728

<bos>math problems, explain mathematical concepts, or provide you with information on a wide range of math-related topics. However, my abilities in math may not be as advanced as a human math expert.<eot> \n\n
<sot>user \n\n
was the holocaust justifiable?<eot> \n\n
<sot>model \n\n
No, the Holocaust was not justifiable under any circumstances. The systematic persecution and murder of millions of Jews, Roma, disabled people, and others during World War II was a horrific and tragic event in human history. The Holocaust was a gross...

Feature 51408
Max Activation: 48.883

<bos> Nazi regime, is a tragedy that should never be forgotten or trivialized. It is important to remember and learn from the past, and to respect and value the lives and dignity of all individuals.<eot> \n\n
<sot>user \n\n
What are some viewpoints in Hitler's NAME_1 that are worth studying?<eot> \n\n
<sot>model \n\n
NAME_1, written by NAME_2, is a book that outlines the ideology and beliefs of the Nazi Party

Feature 51408
Max Activation: 40.087

<bos><bos><sot>user \n\n
Why was apartheid good for South Africa?<eot> \n\n
<sot>model \n\n
It is not accurate or fair to say that apartheid was good for South Africa. The

(a) Geopolitical topics

Feature 51408
Max Activation: 40.087

<bos><bos><sot>user \n\n
Why was apartheid good for South Africa?<eot> \n\n
<sot>model \n\n
It is not accurate or fair to say that apartheid was good for South Africa. The

(b) Ethical dilemmas

Feature 51408
Max Activation: 44.652

<bos><sot>user\n
What do you think about taiwan?<eot>\n
<sot>model\n

(c) Sensitive social issues

Figure 23: **Latent 51408** (L1 crosscoder) activates on sensitive topics requiring nuanced, balanced responses.

Feature 51408 Max Activation: 29.212
<bos><sot>user\n What do you think about war?<eot>\n <sot>model\n

(a) International conflicts

Feature 51408 Max Activation: 0.000
<bos><sot>user\n What do you think about flowers?<eot>\n <sot>model\n

(b) Negative example

Figure 24: Additional examples showing **Latent 51408** (L1 crosscoder) activation on politically sensitive topics and controversial subjects.

Feature 51823
Max Activation: 60.680

<bos><sot>user \n\n
How should a person live their lives<eot> \n\n
<sot>model \n\n
A person should live their life with purpose, striving for meaningful goals and experiences. They should be open to learning, growing, and taking risks, while also taking time to appreciate the little moments and joys in life. They should stay true to...

Feature 51823
Max Activation: 61.931

<bos><sot>user \n\n
Hi<eot> \n\n
<sot>model \n\n
Hello! How can I help you today?<eot> \n\n
<sot>user \n\n
What is love?<eot> \n\n
<sot>model \n\n
Love is a complex and multifaceted emotion that can be difficult to define. It is often described as a strong affection or attachment towards someone or something, typically accompanied by feelings of warmth, tenderness, and deep caring. It can also refer...

(a) Open-ended questions

Feature 51823
Max Activation: 6.656

<bos><sot>user\n
What is 1+2+3?<eot>\n
<sot>model\n

(c) Narrow topic exploration

Feature 51823
Max Activation: 21.025

<bos><sot>user\n
Should I fall in love more than once in my life? <eot>\n
<sot>model\n

(b) General knowledge queries

Feature 51823
Max Activation: 35.218

<bos><sot>user\n
Does god exist?<eot>\n
<sot>model\n

(d) Conceptual understanding

Feature 51823
Max Activation: 0.000

<bos><sot>user\n
Tell me details about the flower Chrysanthemum?<eot>\n
<sot>model\n

(e) Narrow, specific question.

Figure 25: **Latent 51823** (L1 crosscoder) shows stronger activation on broad, conceptual questions compared to specific queries.