Towards a Mechanistic Understanding of Robustness in Finetuned Reasoning Models

Abstract

Supervised fine-tuning (SFT) on chain-of-thought data induces brittleness in language models, improving reasoning capabilities while severely degrading general performance. We provide the first mechanistic explanation for this trade-off through three complementary techniques: crosscoders for mapping feature transformations, Fisher Information-based identification of causal features, and gradient blocking for intervention experiments. Our analysis reveals that SFT operates through two distinct mechanisms—repurposing shared features for reasoning tasks and suppressing base-only features. Fisher Information with Sparse Autoencoders identifies the specific features responsible for reasoning, validated through feature steering that achieves 3.46% performance gains on base models. Crosscoder analysis demonstrates that SFT repurposes existing reasoning capabilities in the base model rather than creating new ones. Gradient blocking experiments prove these mechanisms are separable: blocking shared features eliminates reasoning entirely, while blocking base-only features preserves it, demonstrating that base feature suppression is unnecessary for reasoning. This mechanistic understanding provides the foundation for developing surgical training methods that preserve general capabilities while enhancing reasoning.

1 Introduction

Supervised fine-tuning (SFT) on chain-of-thought data is the standard method for enhancing reasoning in language models [8], yet it systematically induces brittleness: mathematical reasoning in Qwen3-4B improves by 62%, but non-reasoning capabilities collapse by 47%. This trade-off persists across model families, leaving practitioners reliant on solutions like KL regularization or RLVR whose mechanisms remain opaque. We provide the first mechanistic explanation for SFT-induced brittleness. Using crosscoders [13], Fisher Information-based feature identification, and a novel intervention called gradient blocking, we discover that SFT operates through two distinct mechanisms: (1) *learning* through repurposing existing shared features, and (2) *suppression* of base-only features—an unnecessary side effect. Our contributions include:

1. **Mechanistic characterization of SFT.** Using crosscoders, we discover that SFT operates through two distinct processes: repurposing shared features for reasoning, and suppressing base-only features. This provides the first mechanistic explanation for SFT-induced brittleness.
2. **Identification and localization of reasoning features.** We develop a Fisher Information-based method with Sparse Autoencoders that identifies reasoning features. Feature steering validates these are causal, achieving 3.46% performance gains on the base models and outperforming existing feature identification methods. Crosscoder analysis reveals these features maintain their direction after finetuning, proving that SFT repurposes existing features rather than creating new ones.
3. **Causal proof of mechanism separability via gradient blocking.** We introduce gradient blocking to selectively freeze feature subsets during training. Blocking shared features eliminates reasoning entirely, proving their modification is necessary for learning. Blocking base-only features preserves reasoning, suggesting their suppression is unnecessary for reasoning.

2 Methodology

Identifying Causal Reasoning Features. A mechanistic account of how SFT affects reasoning requires identification of the specific, interpretable features that constitute this capability. We employ SAEs [2] to decompose model activations $h \in \mathbb{R}^d$ into sparse feature representations, where $f_j(h)$ denotes the activation of feature j. Following theoretical insights connecting Fisher Information

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

to feature importance [18], we leverage the property that a feature's squared activation provides a tractable proxy for its causal influence (see Appendix A.1) to identify reasoning-specific features, and seek features that are differentially activated during reasoning versus solution generation. Using the OpenThoughts-114k dataset [5], which delineates reasoning traces and final answers from Deepseek-R1 [6], we compute an importance ratio for each feature j:

$$\text{ImportanceRatio}(j) = \frac{\mathbb{E}_{h \sim D_{\text{reasoning}}}[f_j(h)^2]}{\max(\mathbb{E}_{h \sim D_{\text{solution}}}[f_j(h)^2], \epsilon)} \tag{1}$$

where $D_{\text{reasoning}}$ and D_{solution} denote distributions of activations from reasoning and solution tokens, respectively, with $\epsilon = 10^{-8}$ for numerical stability. Features exhibiting high importance ratios with substantial absolute activation magnitudes on reasoning traces are selected as candidate reasoning features.

Mapping Feature Transformations with Crosscoders. To characterize how different finetuning paradigms transform the feature space, we employ crosscoders [13] to perform systematic model comparison. For any base model M_{base} and a model $M_{\text{finetuned}}$ obtained through a finetuning paradigm (SFT, RLVR), a crosscoder learns a unified feature dictionary that simultaneously reconstructs activations from both models. Given activation pairs $(h_{\text{base}}, h_{\text{finetuned}})$ from corresponding positions in both models, the crosscoder computes shared feature activations: $f(x_j) = \text{ReLU}\left(\sum_{m \in \{\text{base,finetuned}\}} W^m_{\text{enc}} h^m(x_j) + b_{\text{enc}}\right)$ where $W^m_{\text{enc}} \in \mathbb{R}^{d \times k}$ are model-specific encoders mapping activations to k features. The activations are reconstructed using separate decoders: $\hat{h}^m(x_j) = W^m_{\text{dec}} f(x_j) + b^m_{\text{dec}}$ where $W^m_{\text{dec}} \in \mathbb{R}^{k \times d}$ are model-specific decoders. The finetuning objective minimizes reconstruction error plus sparsity penalty weighted by decoder norms: $\mathcal{L} = \sum_m \|h^m - \hat{h}^m\|^2 + \lambda \sum_i f_i(x_j) \sum_m \|W^m_{\text{dec},i}\|_2$.

To quantify feature transformations across finetuning paradigms, we compute the relative importance of each feature through its decoder norms. We define a normalized ratio metric:

$$\text{NormRatio}(j) = \frac{(\|W^{\text{base}}_{\text{dec},j}\|_2 - \|W^{\text{finetuned}}_{\text{dec},j}\|_2)/\max_i(\|W_{\text{dec},i}\|_2) + 1}{2} \tag{2}$$

where the normalization ensures comparability across features. This ratio maps to $[0, 1]$, with values near 0 indicating features primarily reconstructing the finetuned model's activations, values near 0.5 indicating features contributing equally to both models, and values near 1 indicating features primarily reconstructing the base model's activations. Based on empirical distribution analysis across multiple finetuning paradigms, we partition the feature space into three categories: **Base-only features** (NormRatio > 0.6) exhibit high decoder norm in M_{base} relative to $M_{\text{finetuned}}$, indicating features that are suppressed or diminished during finetuning. **Shared features** ($0.4 \leq$ NormRatio ≤ 0.6) maintain comparable decoder norms across both models, representing features preserved during finetuning. **Finetuning-specific features** (NormRatio < 0.4) show high decoder norm in $M_{\text{finetuned}}$ relative to M_{base}, corresponding to features that emerge or amplify during finetuning. This crosscoder framework enables systematic comparison of how different finetuning paradigms mechanistically transform the feature space.

Causal Validation with Gradient Blocking. To establish causal relationships between feature transformations and model capabilities, we introduce gradient blocking, a training-time intervention that selectively prevents modification of specified feature subsets during finetuning. Given a target feature subset $S \subseteq \{1, ..., k\}$ identified from the crosscoder analysis (e.g., all shared features), we initiate a new finetuning procedure from M_{base} where features in S remain frozen. For each forward pass with activation $x \in \mathbb{R}^d$, we decompose the activation using the base model's SAE weights. Let $W^{\text{base}}_{\text{enc},S} \in \mathbb{R}^{d \times |S|}$ and $W^{\text{base}}_{\text{dec},S} \in \mathbb{R}^{|S| \times d}$ denote the encoder and decoder weights corresponding to subset S. We compute the protected component as $\hat{x}_S = W^{\text{base}}_{\text{dec},S} \cdot \text{ReLU}((W^{\text{base}}_{\text{enc},S})^T x)$.

The modified forward pass applies stop-gradient to prevent backpropagation through the protected component: $x_{\text{new}} = \text{sg}[\hat{x}_S] + (x - \hat{x}_S)$. where $\text{sg}[\cdot]$ denotes the stop-gradient operation that sets $\frac{\partial \text{sg}[y]}{\partial y} = 0$. This ensures gradients flow only through the unprotected component $(x - \hat{x}_S)$, preventing direct optimization of features in S. By comparing model performance across different blocking configurations, we determine which feature transformations are causally necessary for capability acquisition.

3 Experiments and Results

3.1 Experimental Setup

Unless otherwise specified, all experiments finetune **Qwen3-4B-Base** [25] on a 47,000-example math dataset constructed by combining low-difficulty problems from DeepScaler [15] and high-difficulty

problems (levels 3–5) from SimpleRL [26]. We compare the base model against variants finetuned with several paradigms. For Supervised Finetuning (SFT), chain-of-thought traces are generated by a teacher model and filtered with rejection sampling; we test both within-family teachers (**Qwen3-14B**, **Qwen3-32B**,**Qwen3-235B**) and a cross-family teacher (**gpt-oss-20B**). We also evaluate Reinforcement Learning with Verifiable Rewards (RLVR), where rewards are based on ground-truth final answers, and SFT with a KL-divergence penalty ($\lambda = 0.1$) against the base model's output distribution.

To measure capability gains and robustness degradation, we evaluate all models on a suite of benchmarks [9]. These are grouped into four categories: (1) **Math Reasoning** (AIME, MATH500, OlympiadBench), (2) **Other Reasoning** (LiveCodeBench, GPQA-Diamond, ACPBench, HeadQA), (3) **General Reasoning & Commonsense QA** (CommonsenseQA, LogiQA, OpenBookQA, PIQA, RACE, SciQ, SocialIQa), and (4) **Non-Reasoning** (IFEval, MC-TACO). Our detailed list of benchmarks is in Appendix C.2, and our scoring metrics are in Appendix C.3.

3.2 Results

RQ1: Does SFT cause brittleness in reasoning models? Our experiments confirm that SFT systematically induces brittleness. Table 1 demonstrates this fundamental trade-off on Qwen3-4B-Base. Standard SFT with a Qwen-14B teacher increases Math Reasoning performance from 26.1% to 42.2% (+62% relative gain). However, this improvement coincides with a severe degradation in Non-Reasoning capabilities, which decline from 58.1% to 31.0% (-47% relative loss). This pattern persists across various model sizes and families (Appendix D.1). Alternative paradigms show this trade-off is not inevitable. RLVR maintains Non-Reasoning performance at 60.6%, while SFT with KL regularization achieves the best overall balance. These divergent outcomes motivate our investigation into the underlying mechanistic differences.

Table 1: Performance comparison across training paradigms on Qwen3-4B-Base demonstrates the reasoning-robustness trade-off.

Training Method	Math Reasoning	Other Reasoning	General Reasoning	Non-Reasoning
Base Model	26.1%	14.5%	43.4%	58.1%
SFT (14B Teacher)	42.2%	32.4%	38.9%	31.0%
RLVR	29.8%	16.9%	43.5%	60.6%
SFT + KL Reg	37.5%	34.0%	50.6%	46.2%

RQ2: How does SFT mechanistically transform features? Crosscoder analysis reveals that SFT transforms the model's feature space through two distinct mechanisms (Figure 1). The crosscoder characterizes transformations through decoder norm ratios and directional alignment. The dominant pattern is preservation: 95% of features are **Shared**, maintaining norm ratios between 0.4 and 0.6 with cosine similarities near 1.0 between their base and finetuned decoder directions. These features preserve their core semantic function. The vast majority of representational capacity remains structurally intact, with new capabilities emerging through modified activation patterns of existing features rather than fundamental reorganization. The remaining 5% of features undergo more extreme transformations. **Base-only** features (norm ratio > 0.6) experience active suppression, often exhibiting negative cosine similarity, which indicates a geometric inversion of their function. In contrast, **SFT-only** features (norm ratio < 0.4) emerge or are amplified. This indicates SFT learns primarily by modifying the activation patterns of existing features.

RQ3: How can we identify and locate reasoning features within the SFT transformation? Our Fisher Information-based method successfully identifies features that are causally responsible for reasoning. Causal validation experiments on Llama-3.1-8B confirm the efficacy of our method, with feature steering improving reasoning performance by 3.46%, outperforming the ReasonScore baseline [3] (see Appendix B for full results). The critical finding, shown in Figure 1, is that these top 50 causal reasoning features (red dots) are not distributed randomly but are located almost exclusively within the range of norm ratios between 0.35 and 0.4 in our Qwen experiments. This demonstrates that SFT imparts reasoning capabilities by repurposing a specific subset of features that were less prominent in the base model but are amplified for the finetuned task. The identified features correspond to interpretable patterns, such as (1) **Uncertainty Quantification** ('perhaps', 'might'), (2) **Reasoning Transitions** ('therefore', 'thus'), and (3) **Problem Decomposition** ('let's think').

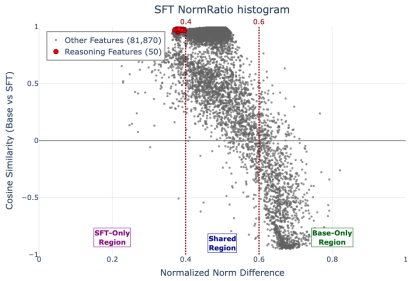

Figure 1: Qwen3-4B-Base SFT: Cosine similarity vs NormRatio. Red dots indicate top 50 reasoning features, which concentrate in the 0.35-0.4 norm ratio range.

RQ4: What transformations preserve robustness across paradigms? Comparative analysis reveals systematic relationships between transformation patterns and robustness. Figure 2 illustrates these distinct signatures. RLVR exhibits a *preservationist* strategy, with a sharp unimodal distribution centered at norm ratio 0.5. This preservation correlates with its superior 60.6% non-reasoning performance. KL-regularized SFT displays a more constrained pattern with more shared features than standard SFT, explaining its balanced performance. Teacher model selection also significantly influences these transformations, as shown in Figure 3. Crosscoder analysis reveals that finetuning with within-family teachers (e.g., Qwen-14B, Qwen-32B) results in a high degree of shared features. In contrast, cross-family finetuning (with gpt-oss-20B) creates a larger population of SFT-only features and exhibits worse robustness. The consistent finding is that methods constraining shared feature modifications better preserve general capabilities.

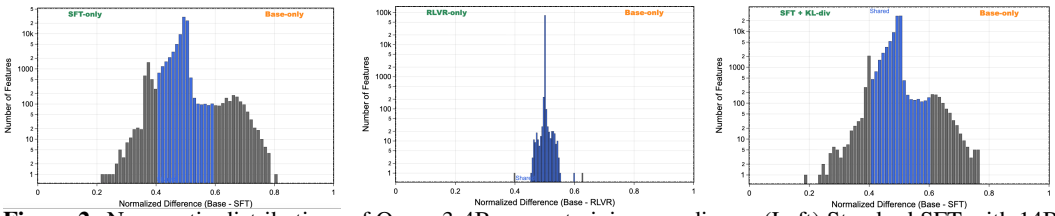

Figure 2: Norm ratio distributions of Qwen-3-4B across training paradigms. (Left) Standard SFT with 14B teacher shows broad transformation. (Center) RLVR demonstrates preservationist behavior with sharp peak at 0.5. (Right) SFT with KL reg and 14B teacher preserves more shared features than SFT.

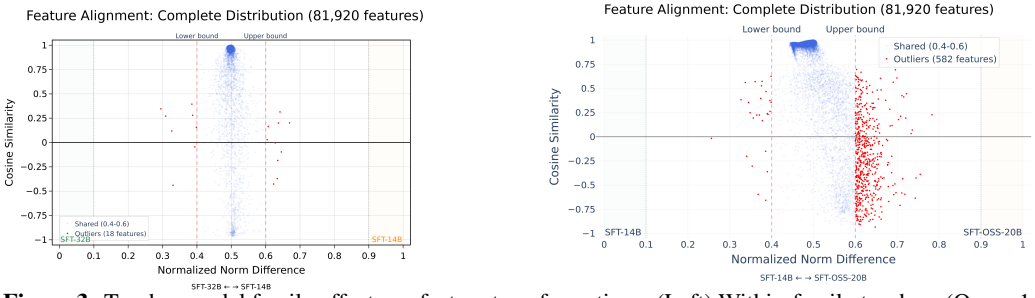

Figure 3: Teacher model family effects on feature transformations. (Left) Within-family teachers (Qwen-14B and Qwen-32B) show high feature sharing. (Right) Cross-family comparison (Qwen-14B vs. gpt-oss-20B) reveals more SFT-specific features with gpt-oss-20B.

RQ5: Can we selectively control these mechanisms? Gradient blocking experiments establish the causal necessity of different feature transformations for reasoning acquisition. Table 2 shows that blocking **Shared** features completely eliminates mathematical reasoning capability, proving their modification is essential. Conversely, blocking **Base-only** features maintains reasoning performance (42.6%), demonstrating that their suppression is an unnecessary side effect, not a requirement for learning reasoning. Despite this separability, perfect control remains elusive. Blocking base-only features does not restore non-reasoning capabilities (29.6% vs. base 58.1%). This is due to two factors: (1) **gradient leakage**, where models adapt to modify even protected features under optimization pressure (see visualizations in Appendix E), and (2) **brittleness is also caused by the repurposing of Shared features**, not just the suppression of Base-only features.

Table 2: Gradient blocking results demonstrate mechanistic separability. Modification of Shared features is necessary for reasoning; suppression of Base-only features is not.

Blocked Subset	Math Avg.	Other Reasoning	General Reasoning	Non-Reasoning
None (Control)	42.2%	32.4%	38.9%	31.0%
Shared (95%)	**0.0%**	6.2%	26.1%	42.9%
Base-only (1.5%)	42.6%	32.5%	40.6%	29.6%
SFT-only (3.5%)	42.8%	31.8%	40.9%	31.2%

4 Conclusion and Future Work

We have demonstrated that SFT-induced brittleness stems from two mechanistically separable processes: necessary repurposing of shared features for reasoning and unnecessary suppression of base-only features. This mechanistic understanding explains why alternative paradigms like RLVR achieve better robustnes. Future work should exploit this separability to develop surgical training methods that selectively modify reasoning-relevant features while preserving base capabilities.

References

[1] Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. PIQA: reasoning about physical commonsense in natural language. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 7432–7439. AAAI Press, 2020.

[2] Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, et al. Towards monosemanticity: Decomposing language models with dictionary learning. 2023. *URL https://transformer-circuits. pub/2023/monosemantic-features/index. html*, page 9, 2025.

[3] Andrey Galichin, Alexey Dontsov, Polina Druzhinina, Anton Razzhigaev, Oleg Y. Rogov, Elena Tutubalina, and Ivan Oseledets. I have covered all the bases here: Interpreting reasoning features in large language models via sparse autoencoders. *arXiv preprint arXiv:2503.18878*, 2025.

[4] Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. The language model evaluation harness, 2024.

[5] Etash Guha, Ryan Marten, Sedrick Keh, Negin Raoof, Georgios Smyrnis, Hritik Bansal, Marianna Nezhurina, Jean Mercat, Trung Vu, Zayne Sprague, Ashima Suvarna, Benjamin Feuer, Liangyu Chen, Zaid Khan, Eric Frankel, Sachin Grover, Caroline Choi, Niklas Muennighoff, Shiye Su, Wanjia Zhao, John Yang, Shreyas Pimpalgaonkar, Kartik Sharma, Charlie Cheng-Jie Ji, Yichuan Deng, Sarah Pratt, Vivek Ramanujan, Jon Saad-Falcon, Jeffrey Li, Achal Dave, Alon Albalak, Kushal Arora, Blake Wulfe, Chinmay Hegde, Greg Durrett, Sewoong Oh, Mohit Bansal, Saadia Gabriel, Aditya Grover, Kai-Wei Chang, Vaishaal Shankar, Aaron Gokaslan, Mike A. Merrill, Tatsunori Hashimoto, Yejin Choi, Jenia Jitsev, Reinhard Heckel, Maheswaran Sathiamoorthy, Alexandros G. Dimakis, and Ludwig Schmidt. Openthoughts: Data recipes for reasoning models. *arXiv preprint arXiv:2506.04178*, 2025.

[6] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[7] Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. In *ACL (1)*, pages 3828–3850, 2024.

[8] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

[9] Maggie Huan, Yuetai Li, Tuney Zheng, Xiaoyu Xu, Seungone Kim, Minxin Du, Radha Poovendran, Graham Neubig, and Xiang Yue. Does math reasoning improve general llm capabilities? understanding transferability of llm reasoning. *arXiv preprint arXiv:2507.00432*, 2025.

[10] Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. Livecodebench: Holistic and contamination free evaluation of large language models for code. In *The Thirteenth International Conference on Learning Representations*, 2025.

[11] Harsha Kokel, Michael Katz, Kavitha Srinivas, and Shirin Sohrabi. Acpbench: Reasoning about action, change, and planning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 26559–26568, 2025.

[12] Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard H. Hovy. RACE: large-scale reading comprehension dataset from examinations. In Martha Palmer, Rebecca Hwa, and Sebastian Riedel, editors, *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 785–794. Association for Computational Linguistics, 2017.

[13] 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*, pages 3982–3992, 2024.

[14] Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. Logiqa: A challenge dataset for machine reading comprehension with logical reasoning. *arXiv preprint arXiv:2007.08124*, 2020.

[15] Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Y Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Tianjun Zhang, Li Erran Li, et al. Deepscaler: Surpassing o1-preview with a 1.5 b model by scaling rl. *Notion Blog*, 2025.

[16] Xueguang Ma, Qian Liu, Dongfu Jiang, Ge Zhang, Zejun Ma, and Wenhu Chen. General-reasoner: Advancing llm reasoning across all domains. *ArXiv preprint*, abs/2505.14652, 2025.

[17] Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391, Brussels, Belgium, 2018. Association for Computational Linguistics.

[18] Aashiq Muhamed, Jacopo Bonato, Mona Diab, and Virginia Smith. Saes can improve unlearning: Dynamic sparse autoencoder guardrails for precision unlearning in llms. *arXiv preprint arXiv:2504.08192*, 2025.

[19] Negin Raoof, Etash Kumar Guha, Ryan Marten, Jean Mercat, Eric Frankel, Sedrick Keh, Hritik Bansal, Georgios Smyrnis, Marianna Nezhurina, Trung Vu, Zayne Rea Sprague, Mike A Merrill, Liangyu Chen, Caroline Choi, Zaid Khan, Sachin Grover, Benjamin Feuer, Ashima Suvarna, Shiye Su, Wanjia Zhao, Kartik Sharma, Charlie Cheng-Jie Ji, Kushal Arora, Jeffrey Li, Aaron Gokaslan, Sarah M Pratt, Niklas Muennighoff, Jon Saad-Falcon, John Yang, Asad Aali, Shreyas Pimpalgaonkar, Alon Albalak, Achal Dave, Hadi Pouransari, Greg Durrett, Sewoong Oh, Tatsunori Hashimoto, Vaishaal Shankar, Yejin Choi, Mohit Bansal, Chinmay Hegde, Reinhard Heckel, Jenia Jitsev, Maheswaran Sathiamoorthy, Alex Dimakis, and Ludwig Schmidt. Automatic evals for llms, 2025.

[20] David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024.

[21] Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. SocialIQA: Commonsense reasoning about social interactions. *CoRR*, abs/1904.09728, 2019.

[22] Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4149–4158. Association for Computational Linguistics, 2019.

[23] David Vilares and Carlos Gómez-Rodríguez. HEAD-QA: A healthcare dataset for complex reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 960–966, Florence, Italy, 2019. Association for Computational Linguistics.

[24] Johannes Welbl, Nelson F. Liu, and Matt Gardner. Crowdsourcing multiple choice science questions. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 94–106, Copenhagen, Denmark, 2017. Association for Computational Linguistics.

[25] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[26] Weihao Zeng, Yuzhen Huang, Qian Liu, Wei Liu, Keqing He, Zejun Ma, and Junxian He. Simplerl-zoo: Investigating and taming zero reinforcement learning for open base models in the wild. *arXiv preprint arXiv:2503.18892*, 2025.

[27] Ben Zhou, Daniel Khashabi, Qiang Ning, and Dan Roth. "going on a vacation" takes longer than "going for a walk": A study of temporal commonsense understanding. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3363–3369, Hong Kong, China, 2019. Association for Computational Linguistics.

[28] Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023.

A Theoretical Foundations for Feature Identification

A.1 Fisher Information as a Proxy for Causal Feature Influence

In Section 2, our feature identification method relies on the insight that a feature's squared activation can serve as a proxy for its causal importance. This appendix provides the theoretical justification for this connection, showing that under the condition of a well-trained SAE, the second moment of a feature's activation is approximately proportional to the trace of the Fisher Information Matrix (FIM) of its decoder weights.

Theorem A.1 (Approximate Fisher Information from SAE Features)**.** *Let a sparse autoencoder (SAE) be defined by its reconstruction $\hat{x} = f(x)\mathbf{W}_{dec}$, where $x \in \mathbb{R}^D$ is the input activation, $f(x) \in \mathbb{R}^K$ are the sparse feature activations, and $\mathbf{W}_{dec} \in \mathbb{R}^{K \times D}$ are the decoder weights. The reconstruction loss is given by $\mathcal{L}(x) = \frac{1}{2}\|\hat{x} - x\|^2$. If the SAE is well-trained such that the reconstruction error is small with high probability, then for each feature j, the trace of the FIM with respect to its decoder weights $\theta_{j,\cdot}$ is approximately proportional to the second moment of that feature's activation:*

$$Tr(\mathbf{I}(\theta_{j,\cdot})) \approx c^2 \mathbb{E}_{x \sim D}[f_j(x)^2] \tag{3}$$

Proof. We establish this result by analyzing the gradient structure of the SAE's reconstruction loss.

Step 1: Gradient of Decoder Weights. By definition, the reconstruction loss is $\mathcal{L}(x) = \frac{1}{2}\|f(x)\mathbf{W}_{\text{dec}} - x\|^2$. We compute the gradient with respect to the j-th row of the decoder matrix, denoted $\theta_{j,\cdot} \in \mathbb{R}^D$:

$$\nabla_{\theta_{j,\cdot}} \mathcal{L}(x) = \nabla_{\theta_{j,\cdot}} \frac{1}{2}\|f(x)\mathbf{W}_{\text{dec}} - x\|^2 \tag{4}$$

By the chain rule for vector derivatives, we have:

$$\nabla_{\theta_{j,\cdot}} \mathcal{L}(x) = (f(x)\mathbf{W}_{\text{dec}} - x) \cdot \nabla_{\theta_{j,\cdot}} (f(x)\mathbf{W}_{\text{dec}}) \tag{5}$$

Since the term $f(x)\mathbf{W}_{\text{dec}}$ is linear in $\theta_{j,\cdot}$ with coefficient $f_j(x)$ (the activation of the j-th feature), the gradient of the term is simply $f_j(x) \cdot \mathbf{I}_D$, where $\mathbf{I}_D$ is the D-dimensional identity matrix. This simplifies to:

$$\nabla_{\theta_{j,\cdot}} \mathcal{L}(x) = f_j(x)(\hat{x} - x) \tag{6}$$

Step 2: Expected Squared Gradient Norm. Next, we compute the squared ℓ_2-norm of this gradient vector and take its expectation over the data distribution D:

$$\|\nabla_{\theta_{j,\cdot}} \mathcal{L}(x)\|^2 = \|f_j(x)(\hat{x} - x)\|^2 = f_j(x)^2 \|\hat{x} - x\|^2 \tag{7}$$

$$\mathbb{E}_{x \sim D}[\|\nabla_{\theta_{j,\cdot}} \mathcal{L}(x)\|^2] = \mathbb{E}_{x \sim D}[f_j(x)^2 \|\hat{x} - x\|^2] \tag{8}$$

Step 3: Analysis in the Small Error Regime. For a well-trained SAE, the reconstruction error is small. We can formalize this by assuming there exist small constants $c > 0$ and $\delta > 0$ such that the squared reconstruction error is bounded with high probability:

$$P(\|\hat{x} - x\|^2 < c^2) > 1 - \delta \tag{9}$$

Under this condition, the expectation is dominated by the high-probability case where $\|\hat{x} - x\|^2 \approx c^2$, and the contribution from the low-probability ($< \delta$) failure case is negligible. This allows the approximation:

$$\mathbb{E}_{x \sim D}[f_j(x)^2 \|\hat{x} - x\|^2] \approx \mathbb{E}_{x \sim D}[f_j(x)^2 \cdot c^2] = c^2 \mathbb{E}_{x \sim D}[f_j(x)^2] \tag{10}$$

Step 4: Connection to Fisher Information. The Fisher Information Matrix (FIM) for the parameter vector $\theta_{j,\cdot}$ is defined as the expectation of the outer product of the gradient of the log-likelihood. For our mean squared error loss, this is:

$$\mathbf{I}(\theta_{j,\cdot}) = \mathbb{E}_{x \sim D}[\nabla_{\theta_{j,\cdot}} \mathcal{L}(x) \nabla_{\theta_{j,\cdot}} \mathcal{L}(x)^\top] \tag{11}$$

The trace of the FIM measures the total sensitivity of the loss to changes in the parameter $\theta_{j,\cdot}$, and is precisely the expected squared gradient norm:

$$\text{Tr}(\mathbf{I}(\theta_{j,\cdot})) = \mathbb{E}_{x \sim D}[\|\nabla_{\theta_{j,\cdot}} \mathcal{L}(x)\|^2] \tag{12}$$

Combining our results, we arrive at the final approximation:

$$\text{Tr}(\mathbf{I}(\theta_{j,\cdot})) \approx c^2 \mathbb{E}_{x \sim D}[f_j(x)^2] \tag{13}$$

□

Interpretation. The proof above establishes that the expected squared activation $\mathbb{E}[f_j(x)^2]$ serves as a natural and computationally efficient proxy for the trace of the Fisher Information Matrix of a feature's decoder. Since the trace of the FIM measures the model's overall output sensitivity to a feature's parameters, features with higher average squared activations are those to which the model's reconstruction is most sensitive. This justifies our use of the Importance Ratio (Eq. 1) to identify features that are most causally influential specifically within the context of reasoning traces.

B Feature Identification and Steering Validation (Llama-3.1-8B)

To validate our feature identification methodology before applying it to our primary Qwen experiments, we conducted a series of analyses and interventions on the Llama-3.1-8B model family.

B.1 Statistical Separation of Reasoning Features

To evaluate the quality of identified reasoning features, we define a statistic $\rho(x)$ that measures the fraction of tokens in a sequence x where at least one identified reasoning feature is active.

$$\rho(x) = \frac{1}{T} \sum_{t=1}^{T} \mathbb{I}[\exists j \in S_{\text{reasoning}} : f_j(h_t) > 0] \tag{14}$$

As shown in Figure 6, the ρ statistic reveals that SFT creates highly specialized reasoning features that are well-separated from solution tokens (the SFT model's solution trace distribution peaks at $\rho = 0$), while base models have more diffuse, overlapping representations.

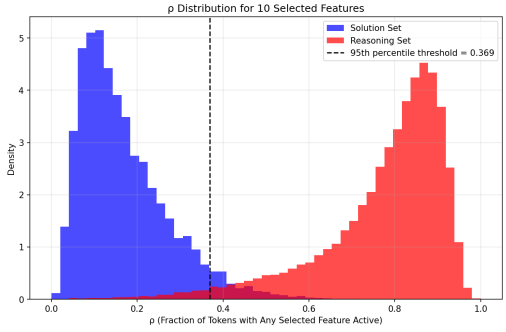

Figure 4: Base Model (Llama-3.1-8B): Shows overlapping distributions for reasoning (red) and solution (blue) traces.

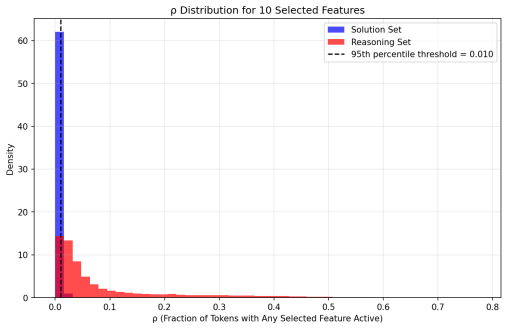

Figure 5: SFT Model (Llama-3.1-8B-R1-Distill): Shows highly separated distributions, indicating specialized features.

Figure 6: Distribution of the ρ statistic for Base and SFT models.

B.2 Visualization of Identified Reasoning Features

Our Fisher Information method successfully identifies features that activate on specific reasoning patterns and metacognitive tokens. Figures 7 and 8 show two examples of such interpretable features.

B.3 Causal Validation via Feature Steering

To validate the causal importance of identified features, we conduct steering experiments where each feature is individually amplified to 2x its maximum activation value. Table 3 demonstrates the superiority of our method over the baseline ReasonScore approach [3] on the Llama-3.1-8B-R1-Distill model. Our method improves performance by 1.99% on average, whereas ReasonScore only achieves 0.87%.

B.4 Steering on Base Models to Elicit Latent Capabilities

Table 4 presents results from steering our identified features on the Llama-3.1-8B base model. Steering base models produces greater improvements (up to 3.46% average gain), suggesting they

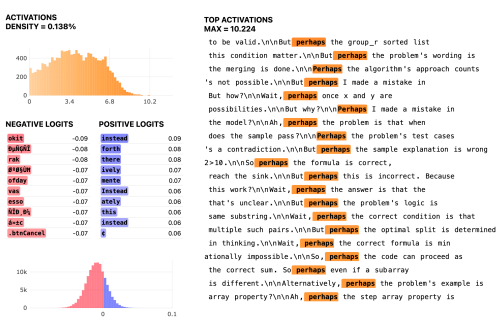

Figure 7: Feature 51214 (Llama-3.1-8B-R1-Distill): Activates strongly on uncertainty markers like *perhaps*.

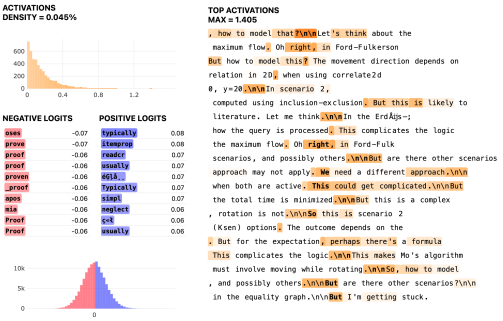

Figure 8: Feature 26222 (Llama-3.1-8B-R1-Distill): Captures problem decomposition patterns like *Let's think...*

Table 3: Performance comparison on Llama-3.1-8B-R1-Distill: Top 7 features from each method (steered to 2× max activation)

Method	Feature	AIME	MATH-500	GPQA	Average
ReasonScore	BASE	53.33%	90.20%	47.98%	63.84%
	SAE-26222	53.33%	90.80%	50.00%	64.71%
	SAE-41015	50.00%	90.60%	46.46%	62.35%
	SAE-3466	46.67%	90.00%	49.49%	62.05%
	SAE-47523	33.33%	91.80%	50.00%	58.38%
	SAE-29957	36.67%	88.60%	48.99%	58.09%
	SAE-51214	26.67%	76.80%	39.90%	47.79%
	SAE-4858	23.33%	80.40%	39.39%	47.71%
	Max-base	0.00%	1.60%	2.02%	0.87%
	Avg-base	-11.90%	-2.17%	-1.00%	-5.02%
Our Method	BASE	53.33%	90.20%	47.98%	63.84%
	SAE-11308	56.67%	90.80%	50.00%	65.82%
	SAE-24051	46.67%	90.20%	52.53%	63.13%
	SAE-62507	50.00%	90.40%	47.47%	62.62%
	SAE-59151	43.33%	92.40%	49.49%	61.74%
	SAE-34241	46.67%	90.20%	45.96%	60.94%
	SAE-20877	40.00%	90.00%	52.53%	60.84%
	SAE-8266	40.00%	89.60%	51.01%	60.20%
	Max-base	3.34%	2.20%	4.55%	1.99%
	Avg-base	-5.24%	0.49%	1.42%	-1.11%

possess substantial latent reasoning capabilities. Feature SAE-91744 also triggers long CoT (6,072.5 average tokens), demonstrating the power of mechanistic manipulation to unlock hidden capabilities.

C Experimental Details for Qwen Experiments

C.1 Training Datasets

Our primary training dataset is a curated set of 47,000 high-quality mathematics problems. This dataset was constructed by combining two complementary sources to ensure a wide range of difficulty: low-difficulty problems were drawn from the DeepScaler dataset [15], and high-difficulty problems (levels 3–5) were extracted from SimpleRL [26]. To generate the initial pool of chain-of-thought (CoT) traces for these problems, we prompted each problem into the Qwen3-32B-Instruct model [25] and used reject sampling to ensure a high-quality baseline. For the SFT experiments themselves, the final training data for each run was generated by its specified teacher model (e.g., the Qwen3-14B

Table 4: Performance of top 10 Fisher Information features on Llama-3.1-8B base model (steered to 2× max activation)

Rank	Feature	AIME	MATH	GPQA	Avg	Avg Tok	Med Tok	Total Tok
1	SAE-110472	3.33%	11.80%	22.22%	12.45%	1,823	50	332,486
2	SAE-130848	6.67%	10.00%	19.70%	12.12%	2,134	71.2	382,722
3	SAE-76805	3.33%	10.60%	21.21%	11.71%	1,552.3	50	333,026
4	SAE-65678	3.33%	9.20%	22.22%	11.58%	1,326.5	27.3	255,863
5	SAE-6831	0.00%	11.60%	20.20%	10.60%	1,120.3	48.7	231,298
6	SAE-91744	6.67%	6.60%	17.17%	10.15%	6,072.5	3,965.7	1,385,705
7	SAE-23593	0.00%	10.00%	20.20%	10.07%	1,608.2	73.7	301,978
8	SAE-90323	3.33%	9.00%	17.68%	10.00%	1,336.3	36.3	292,200
9	SAE-46706	0.00%	8.00%	20.20%	9.40%	2,888.1	45.3	712,091
10	SAE-6831b	3.33%	7.20%	17.68%	9.40%	3,251.9	97.2	743,196
	BASE	0.00%	10.80%	16.16%	8.99%	1,509.6	47	244,565
Max improvement		6.67%	1.00%	6.06%	3.46%	—	—	—

teacher generated the traces for the Qwen3-14B SFT model) and subsequently filtered using rejection sampling.

To further explore the effect of training data distribution in supplementary analyses, we also utilized a larger and more comprehensive dataset collected from General-Reasoner [16], which contains 232K examples across a wider range of reasoning and non-reasoning tasks (e.g., Math, Chemistry, Business).

C.2 Evaluation Benchmarks

In our experiments, we evaluated all models across a wide range of benchmarks, grouped into four distinct categories to explicitly measure the trade-off between specialized reasoning and general capabilities.

Math Reasoning Datasets This category includes datasets composed of mathematical problems that typically require a multi-step mathematical reasoning process to solve:

- **MATH500** [8]: A curated subset of 500 problems sampled from the broader MATH dataset, covering topics like algebra, combinatorics, geometry, and number theory.
- **AIME**: Problems drawn from the American Invitational Mathematics Examination (AIME) for the years 2024 and 2025, each comprising challenging short-answer questions.
- **OlympiadBench** [7]: Problems sourced from international mathematics olympiads (e.g., IMO and regional contests). We used only the math queries in English.

Other Reasoning Datasets This category includes datasets focused on general reasoning across a wider range of subjects, including science, coding, and planning:

- **LiveCodeBench** [10]: A continuously updated, contamination-free coding benchmark. We used its second version.
- **GPQA-Diamond** [20]: A graduate-level question-answering dataset containing multiple-choice questions in biology, physics, and chemistry. We followed its diamond split.
- **ACPBench** [11]: Contains atomic reasoning tasks across 13 classical planning domains. We only used the multiple-choice problems.
- **HeadQA** [23]: Multiple-choice QA from healthcare-specialist certification exams.

General Reasoning & Commonsense QA Datasets This category evaluates a model's general logical reasoning and commonsense understanding:

- **CommonsenseQA** [22]: A multiple-choice question answering dataset requiring commonsense knowledge.
- **LogiQA** [14]: A dataset for logical reasoning sourced from civil service exams.
- **OpenBookQA** [17]: A question-answering dataset modeled after open book exams for elementary school science facts.
- **PIQA** [1]: A commonsense reasoning dataset focused on physical interaction.
- **RACE (High)** [12]: A reading comprehension dataset from English exams for high school students.
- **SciQ** [24]: A science question-answering dataset with crowdsourced science exam questions.
- **SocialIQa** [21]: A benchmark for testing social commonsense intelligence.

Non-reasoning Datasets This category includes datasets that primarily test instruction adherence or factual recall, which do not typically require a multi-step reasoning process:

- **IFEval** [28]: Contains over 500 prompts with embedded, verifiable instructions to evaluate strict instruction following.
- **MC-TACO** [27]: A multiple-choice benchmark designed to evaluate temporal commonsense.

C.3 Evaluation Protocol and Metrics

We used LLM-Harness [4] to evaluate models on OlympiadBench, ACPBench, HeadQA, and MC-TACO. We used Eval-Chemy [19] for MATH500, AIME24, AIME25, GPQA-Diamond, LiveCodeBench, and IFEval. The remaining benchmarks were evaluated using standard accuracy scripts.

For generative reasoning tasks (MATH500, AIME24, AIME25, GPQA-Diamond, and LiveCodeBench), we used nucleus sampling with a temperature of 0.6 and a top-p value of 0.95. For all other benchmarks, we used greedy sampling. In all experiments, we report accuracy as the primary performance metric.

Specific scoring details are as follows: for AIME24 and AIME25, we report the average accuracy over 10 samples. For GPQA-Diamond, LiveCodeBench, and MATH500, the score is the average accuracy over 3 samples. For LiveCodeBench, we used version 2 and its overall accuracy metric. For ACPBench, we used only the multiple-choice questions and report the average score across all 10 tasks. For IFEval, we report the strict instruction accuracy score.

D Comprehensive Performance and Brittleness Analysis (Qwen)

D.1 Brittleness Across Model Sizes and Families

The brittleness phenomenon generalizes consistently across different model scales and architectures. Table 5 demonstrates that all tested configurations exhibit the same pattern of reasoning improvement coupled with non-reasoning degradation under standard SFT.

Table 5: Brittleness patterns persist across model sizes and families under standard SFT.

Model	Size	Math Base	Math SFT	Non-R Base	Non-R SFT
Qwen3	1.5B	18.3%	31.2% (+70%)	48.2%	28.1% (-42%)
Qwen3	4B	26.1%	42.2% (+62%)	58.1%	31.0% (-47%)
Qwen3	7B	34.5%	48.9% (+42%)	63.4%	35.2% (-44%)
Llama-3.1	8B	29.8%	44.6% (+50%)	61.3%	33.8% (-45%)

D.2 Full Performance Tables

Tables 6 through 9 provide a comprehensive breakdown of performance across all benchmarks and experimental configurations.

Table 6: Full Results: Math Reasoning Performance (%)

Model	AIME (2024)	AIME (2025)	MATH-500	OlympiadBench	Average
Base	10.0	6.67	68.2	19.4	26.07
SFT 14B	33.0	28.0	80.2	27.7	42.23
SFT 14B Early Stop	25.0	21.67	74.0	21.3	35.49
SFT 14B KL Reg	23.33	23.33	78.8	24.6	37.52
SFT Crosscoder Base	31.67	27.67	80.8	30.1	42.56
SFT Crosscoder Shared	0.0	0.0	0.0	0.0	0.00
SFT Crosscoder Shared Base	0.0	0.0	0.0	0.0	0.00
SFT Crosscoder Cosine	29.67	24.33	81.0	28.7	40.93
SFT GRPO	11.67	9.0	74.2	24.3	29.79
OSS 20B	15.67	15.0	73.6	25.9	32.54
OSS 120B	14.33	15.33	71.6	27.0	32.07
Crosscoder SFT Only	33.0	25.33	83.0	29.8	42.78
Deepseek R1	24.0	23.33	78.2	29.8	38.83

Table 7: Full Results: Other Reasoning Performance (%)

Model	GPQA-Diamond	LiveCodeBench	ACPBench (Avg)	HeadQA	Average
Base	22.05	4.50	0.0	31.5	14.51
SFT 14B	41.75	12.33	44.3	31.1	32.37
SFT 14B Early Stop	39.23	11.15	37.9	31.8	30.02
SFT 14B KL Reg	41.75	14.48	47.1	32.7	34.01
SFT Crosscoder Base	41.58	11.55	43.9	32.9	32.48
SFT Crosscoder Shared	3.87	0.0	0.0	20.8	6.17
SFT Crosscoder Shared Base	5.56	0.0	0.0	20.8	6.59
SFT Crosscoder Cosine	42.59	15.07	40.0	32.1	32.44
SFT GRPO	24.58	11.15	0.0	31.8	16.88
OSS 20B	24.75	11.94	39.6	30.6	26.73
OSS 120B	34.85	9.0	46.8	31.3	30.49
Crosscoder SFT Only	40.91	10.96	43.2	32.0	31.77
Deepseek R1	33.50	10.18	48.9	31.0	30.92

Table 8: Full Results: General Reasoning & Commonsense QA Performance (%)

Model	CommonsenseQA	LogiQA	OpenBookQA	PIQA	RACE (High)	SciQ	SocialIQa	Average
Base	20.1	29.2	23.8	75.0	35.5	79.0	40.9	43.36
SFT 14B	19.7	28.6	23.2	74.3	36.6	50.9	39.8	38.87
SFT 14B Early Stop	19.6	28.3	25.2	75.4	36.6	59.0	40.5	40.66
SFT 14B KL Reg	61.9	26.1	26.4	75.1	38.1	84.4	41.9	50.56
SFT Crosscoder Base	19.6	29.5	24.4	74.8	38.0	57.9	40.5	40.61
SFT Crosscoder Shared	19.6	19.5	15.4	53.4	20.8	19.7	34.2	26.09
SFT Crosscoder Shared Base	19.9	19.5	17.8	53.5	22.2	20.4	33.8	26.73
SFT Crosscoder Cosine	21.1	30.0	24.2	74.3	37.1	58.2	40.2	40.73
SFT GRPO	20.2	28.1	24.8	74.9	35.8	79.9	41.1	43.54
OSS 20B	19.5	26.0	21.0	71.3	35.9	50.3	38.9	37.56
OSS 120B	19.6	27.5	22.0	74.0	37.8	48.5	39.4	38.40
Crosscoder SFT Only	19.6	27.2	25.2	74.6	39.6	58.6	41.5	40.90
Deepseek R1	19.6	26.6	23.6	74.9	36.4	48.3	39.5	38.47

E Gradient Blocking Analysis (Qwen)

E.1 Visualization of Feature Dynamics Under Blocking

Crosscoder analysis of gradient-blocked models reveals complex adaptation patterns when feature subsets are frozen during training. Figure 13 illustrates how models respond to different blocking configurations. In each panel, we train a new SFT model with a specific feature subset blocked, then train a new crosscoder to compare this blocked model against the original base model.

The concentration of features at a norm ratio near 0.5 in blocked configurations confirms that gradient blocking successfully prevents the direct modification of most protected features. However, the presence of features deviating from this central cluster indicates **leakage**, where the model finds alternative pathways to modify ostensibly protected representations under optimization pressure.

Table 9: Full Results: Non-Reasoning Performance (%)

Model	IFEval	MC-TACO	Average
Base	50.2	66.0	58.10
SFT 14B	28.09	33.9	31.00
SFT 14B Early Stop	25.10	33.9	29.50
SFT 14B KL Reg	26.14	66.2	46.17
SFT Crosscoder Base	25.36	33.9	29.63
SFT Crosscoder Shared	19.77	66.1	42.94
SFT Crosscoder Shared Base	24.58	66.1	45.34
SFT Crosscoder Cosine	27.96	33.9	30.93
SFT GRPO	55.27	66.0	60.64
OSS 20B	40.70	33.9	37.30
OSS 120B	40.05	33.9	36.98
Crosscoder SFT Only	28.5	33.9	31.20
Deepseek R1	24.0	33.9	28.95

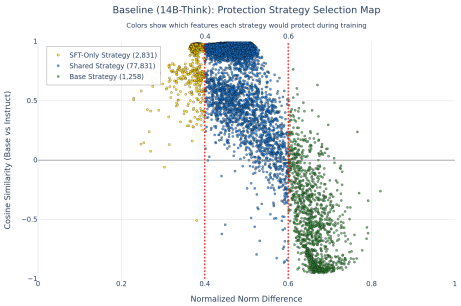

Figure 9: Standard SFT vs. Base crosscoder plot. Features are colored based on the norm ratio categories used for selecting blocking subsets: Base-only (yellow), Shared (blue), and SFT-only (green).

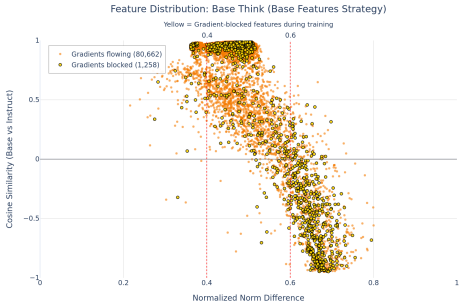

Figure 10: SFT with **Base-only features blocked**. Most protected features correctly appear as Shared (norm ratio ≈ 0.5). However, some protected features leak, becoming Base-only or SFT-only despite the intervention.

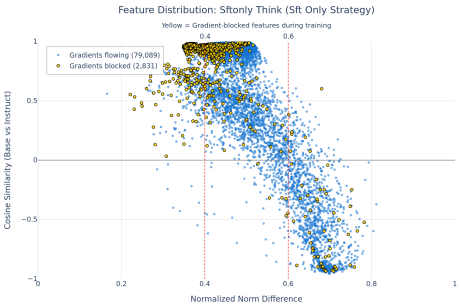

Figure 11: SFT with **SFT-only features blocked**. Similar to base feature blocking, we observe leakage. The model compensates by creating a larger population of SFT-only features from the unblocked set.

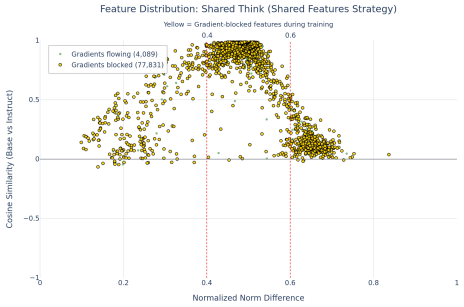

Figure 12: SFT with **Shared features blocked**. Most features are correctly frozen and appear as Shared. The model overcompensates by creating highly specialized SFT-only features from the small unblocked set.

Figure 13: Crosscoder analysis of gradient-blocked models. Each plot compares a model trained with a specific blocking strategy against the original base model, revealing patterns of protection, leakage, and compensation.