Optimized Semantic Steering via Sparse Autoencoder Adapters

Alexander R. Farhang[1] **Anne L. Erickson**[1] **Yisong Yue**[1]

[1]California Institute of Technology
afarhang@caltech.edu

Abstract

We study the problem of steering LLMs, where the goal is to intervene on hidden layer activations in order to improve specific behavioral properties. We are particularly interested in settings where one can optimize over a large candidate set of steering interventions to find valuable, task-related perturbations. We take the approach of using sparse autoencoder adapters coupled with natural language feature descriptions to identify disentangled latent dimensions, after which we select and optimize a subset of the the latent codes that are relevant for the target downstream behavior. A key benefit of our approach is the ability to leverage LLM priors to guide feature selection without manual inspection of relevant features. We empirically demonstrate that our approach can generate steered LLM variants that outperform unsteered LLMs on natural language tasks. Furthermore, the selected steering variants can exhibit cross-lingual transfer, providing task improvements on other languages, unseen during selection or tuning. Our method enables tractable, optimized LLM steering by decomposing the problem into discrete feature selection and continuous optimization. This work demonstrates how tools from mechanistic interpretability can be leveraged to improve model capabilities.

1 Introduction

Large language models (LLMs) have rapidly improved their general problem solving ability in recent years, due in large part to massive scaling of train-time compute for pre-training [11]. However, to alter LLM behavior and improve capabilities for specific downstream tasks, multiple techniques can be used, including the modification of weights, context, and activations. Weights are typically modified through supervised finetuning [30], reinforcement learning [7], and other post-training approaches as well as direct model editing [15]. LLM performance can be altered through modifications to the context and produced tokens, including Chain of Thought [26], test-time scaling [16], and agentic harnessing [29] and tool use [21]. Finally, direct perturbation of neural activations has emerged as a promising method to control LLM response properties, including activation engineering techniques [24] like model steering [5] and activation patching [10]. Direct intervention on model activations can be challenging, as they are extremely high-dimensional spaces and semantically dense. Common approaches include finding steering vectors by computing the difference of mean activations between sets of contrastive example pairs or by using the weight vectors of trained linear classification probes [1, 13, 17, 22, 24]. Steering LLMs has typically involved intervening on identified concepts, styles, or even reasoning techniques, and ensuring the outputs reference and use the topics or strategies [12, 25, 27, 28]. Recent work has proposed using steering methods during training to prevent the development of particular unwanted features like sycophancy [5].

Sparse Autoencoders (SAEs) have been introduced as a method in mechanistic interpretability research with the goal of crisply decomposing neural activations into individual, understandable

Preprint.

concepts [4, 6, 23]. Neural activations are typically highly polysemantic—individual neurons are often activated in response to different concepts, a phenomenon also noted in biological brains and termed "mixed selectivity" [20]. Recent work has combined steering with SAEs, by perturbing in the higher-dimensional, but more human-interpretable SAE latent space [14, 23].

Our approach continues on this thread, performing interpretable, sparse interventions in the SAE latent space rather than dense modifications in the raw activation space. We leverage the intelligence of frontier LLMs to define problem solving strategies to be utilized by a smaller steerable model, performing semantic search over labeled SAE latent features to select for feature vectors most closely related to conceptual, metacognitive strategies to solve downstream tasks. Although our interventions are compound (combining multiple interpretable features), the SAE space provides a more traversable landscape for steering compared to random neurons or the underlying LLM activation space, where individual dimensions lack clear semantic meaning. Through this combination of approaches, we innovate a method to more tractably maneuver the enormous size of the steering space by decomposing the problem into discrete feature selection and continuous optimization.

2 Methodology for Optimized Semantic Steering

SAEs encode high-dimensional, but disentangled latent spaces. As the features recovered should be monosemantic, it can be expected that finely-grained steering optimization could be performed if the right subset of dimensions to optimize over is chosen appropriately. To address this, we propose a method for optimized semantic steering, augmenting an LLM with three modules: a Model Adapter, a Feature Selector, and an Optimizer (Fig. 1). The Model Adapter defines an interpretable latent space and provides a mapping from the perturbation space to the LLM's activation space, enabling steered inference from a modified latent representation, $\widetilde{\boldsymbol{h}}$. The Feature Selector generates the feature set $\mathbf{V}$, which defines a perturbation subspace in the Model Adapter's latent space. The Optimizer selects the feature weight vector, $\boldsymbol{\alpha}$, scaling each component of the feature set for model steering: $\widetilde{\boldsymbol{h}} = \boldsymbol{h} + \boldsymbol{\alpha} \odot \boldsymbol{V}$.

Each feature v_i corresponds to a one-hot vector $\mathbb{1}\{v_i\}$ in the Model Adapter latent space that is 1 at the dimension corresponding to feature v_i and 0 elsewhere. Model Adapter latent features are modified by applying the tuned weights: $\widetilde{\boldsymbol{h}} = \boldsymbol{h} + \sum_i \alpha_i \cdot \mathbb{1}\{v_i\}$, where $\boldsymbol{h}$ is the original encoding and α_i is the learned scaling weight for feature v_i. The Model Adapter decodes the steered latent back to LLM activation space and the modified activation continues through the LLM's forward pass.

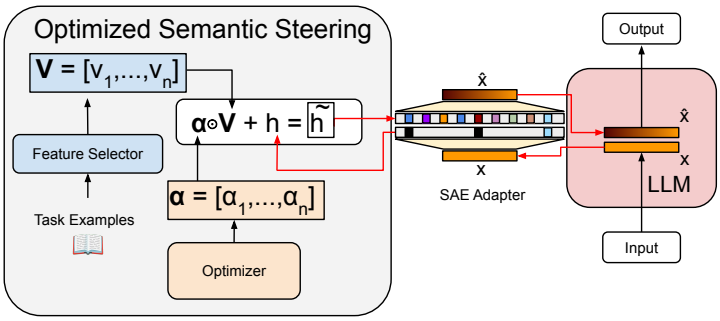

Figure 1: Diagram of our Optimized Semantic Steering method.

2.1 Model Adapter

Our Model Adapter is an SAE, coupled with natural text description of features, trained to reconstruct residual hidden layer activations, $x \in \mathbb{R}^d$, at a specific layer in our steerable LLM:

$$\hat{x} = W_{dec}h + b_{dec}, \quad h = \sigma(W_{enc}x + b_{enc})$$

where $W_{enc} \in \mathbb{R}^{m \times d}$, $b_{enc} \in \mathbb{R}^m$, $W_{dec} \in \mathbb{R}^{d \times m}$, and $b_{dec} \in \mathbb{R}^d$ are the learned parameters. The nonlinearity, $\sigma(\cdot)$, is the ReLU activation function; other approaches to training SAEs use different activation functions to promote sparsity, including Jump-ReLU [19], Gated [18], TopK [8].

To encourage sparsity, the latent dimension of the SAE is set to be much wider than the dimension of the neural network activations at that layer, d. Additionally, the SAEs are typically trained with a reconstruction loss regularized with a sparsity constraint, in this case L_1 regularization:

$$L(x, \hat{x}, h) = ||x - \hat{x}||_2^2 + \lambda||h||_1$$

To steer model behavior from this SAE latent space, the error term of the lossy SAE is calculated, $\epsilon = x - \hat{x}$, and intervention is performed by modifying the desired features in the SAE latent, h, decoding, and re-adding the error term:

$$\widetilde{x} = W_{dec}\widetilde{h} + b_{dec} + \epsilon.$$

2.2 Feature Selector

We set the Feature Selector to be a powerful LLM combined with a semantic feature search. Natural language-associated SAE features are selected by Claude Sonnet 4: we prompt the LLM with 24 set-aside in-context examples, as well as instruction to generate N non-overlapping strategic concepts $\mathbf{C} = \{c_1, c_2, ..., c_N\}$ that should be used to solve questions of this type (see A.1.1). We then use semantic feature search to identify the top-k most similar SAE features per concept, c_i, yielding a feature set $\mathbf{V} = \{v_1, v_2, ..., v_{N \cdot k}\}$. For all experiments, we generate $N = 10$ concepts and $k = 2$ features per concept, resulting in a steering vector of 20 features (using the Ember SDK [3] for similarity querying) per dataset that remain fixed for optimization.

2.2.1 Feature Selector Ablation - Random Features

We substitute our Feature Selector with a random choice of 20 SAE features to perform optimization on, ignoring the semantically rich feature label associated with each. This serves as the primary baseline to demonstrate the value of semantic selection over arbitrary feature steering.

2.3 Optimizer

For our optimizer, we use out-of-the-box Bayesian Optimization using Tree-Structured Parzen Estimation, which tunes the features, $\mathbf{V}$, to find optimal steering weights $\boldsymbol{\alpha} = \{\alpha_1, \alpha_2, ..., \alpha_m\}$, where $\alpha_i \in [-0.3, 0.3]$ for each feature $v_i \in \mathbf{V}$. Optimization directly maximizes accuracy on a train set of 500 examples, running 100 trials, with 4 parallel workers. We implement median pruning (checking every 50 samples after the first 100) and early stopping for catastrophic trials (accuracy below 5% on the first 100 samples). We select the top 3 performing variants on our train set and observe performance on a validation and test set.

2.4 Data and Models

We use a subset of the multilingual datasets provided in the BUFFET benchmark, namely XWinograd (commonsense reasoning), XNLI (natural language inference), and Amazon Reviews (sentiment classification) [2]. By default, we use the English subsets, except for the language transfer experiments, where we investigate the Xwinograd Portuguese, Japanese, and Chinese datasets.

We use the lightweight open source Llama-3.1-8B-Instruct model [9] and the open source SAE, Llama-3.1-8B-Instruct-SAE-l19 released by Goodfire [14], which is associated with text features for every SAE hidden dimension neuron ($65,536$ neurons), before filtering for toxicity. This SAE is trained using the LMSYS-Chat-1M dataset [31] on the residual stream at layer 19 of Llama-3.1-8B, achieving an L0 count of 91 (mean number of activated features). Llama-3.1-8 is run with a maximum output token limit of 500, temperature of 0.6, and top-p of 0.9. A maximum output token limit is critical, as Llama-3.1-8B in its default case will often repeat sentences unnecessarily. In this work, we treat our steered LLM as a "reasoning engine" and do not require multiple choice formatting, instead feeding its output and the multiple choice mapping to a separate formatter LLM, GPT-4.1-mini with temperature 0.7 (see A.1.3).

3 Results

We find that our approach robustly finds model variants with improved downstream performance on the commonsense reasoning tasks in the Xwinograd English dataset. The LLM Feature Selector chose strategic concepts that match to SAE features that can be steered to generalize much better than random features (Fig. 2, Fig. 4). On the natural language logical inference task, XNLI, our approach selects $\frac{2}{3}$ variants that outperform the baseline LLM with SAE. However, we find small improvement on the sentiment classification task, Amazon reviews, likely due to task performance saturation: Llama-3.1-8B already performs within a percent of GPT-4.1-mini, which far outperforms the Llama model on other tasks (Table 1 in A).

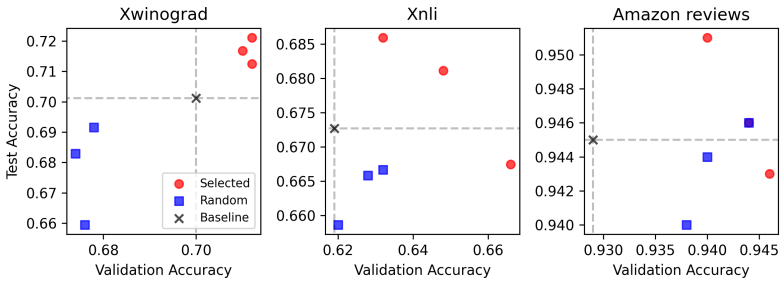

Figure 2: Generalization performance of Optimized Semantic Steering variants vs steering of randomly selected SAE features.

3.1 Semantic steering can generalize across language

We dive deeper into the steered model variants selected by our method on the XWinograd dataset. As this is a multilingual dataset, we apply the identical steered model variants that were selected exclusively on the English data subset to other languages, in a language transfer experiment. We find that they transfer extremely well to Portuguese, with some benefit on Japanese and Chinese (Fig. 3), suggesting that the features selected improve reasoning in a shared way across languages.

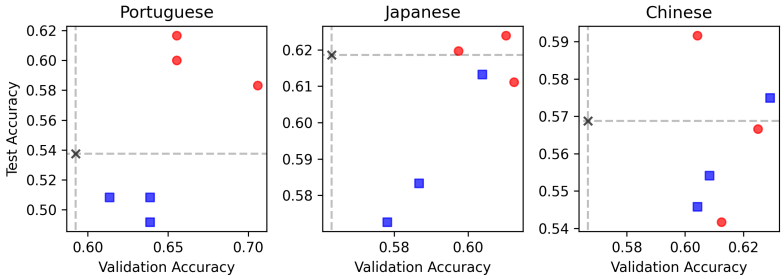

Figure 3: Language transfer of steered model variants tuned only on English Xwinograd questions.

4 Limitations

While this work serves as a proof of concept for optimized semantically-guided steering for downstream task performance, it is important to note that it only uses a single LLM (8B) and SAE. Appropriate granularity of SAE feature labels is key for this approach and understanding the steerability of features labeled with different auto-labeling methods would be informative. Furthermore, Bayesian Optimization is computationally expensive; extensions could more efficient feature selection methods, including those that incorporate feature selection iteratively. Future work should explore larger models, more diverse and challenging datasets, as well as multilayer inventions.

References

[1] Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. Refusal in language models is mediated by a single direction, 2024. URL https://arxiv.org/abs/2406.11717.

[2] Akari Asai, Sneha Kudugunta, Xinyan Velocity Yu, Terra Blevins, Hila Gonen, Machel Reid, Yulia Tsvetkov, Sebastian Ruder, and Hannaneh Hajishirzi. Buffet: Benchmarking large language models for few-shot cross-lingual transfer, 2023. URL https://arxiv.org/abs/2305.14857.

[3] Daniel Balsam, Myra Deng, Nam Nguyen, Liv Gorton, Thariq Shihipar, Eric Ho, and Thomas McGrath. Goodfire ember: Scaling interpretability for frontier model alignment. Goodfire Research Blog, Dec 2024. URL https://www.goodfire.ai/blog/announcing-goodfire-ember. Online; accessed 2025-08-21.

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

[5] Runjin Chen, Andy Arditi, Henry Sleight, Owain Evans, and Jack Lindsey. Persona vectors: Monitoring and controlling character traits in language models, 2025. URL https://arxiv.org/abs/2507.21509.

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

[7] DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, and Junxiao Song et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

[8] Leo Gao, Tom Dupré la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. Scaling and evaluating sparse autoencoders, 2024. URL https://arxiv.org/abs/2406.04093.

[9] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, and Abhishek Kadian et al. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

[10] Stefan Heimersheim and Neel Nanda. How to use and interpret activation patching, 2024. URL https://arxiv.org/abs/2404.15255.

[11] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. Training compute-optimal large language models, 2022. URL https://arxiv.org/abs/2203.15556.

[12] Kai Konen, Sophie Jentzsch, Diaoulé Diallo, Peer Schütt, Oliver Bensch, Roxanne El Baff, Dominik Opitz, and Tobias Hecking. Style vectors for steering generative large language model, 2024. URL https://arxiv.org/abs/2402.01618.

[13] Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model, 2024. URL https://arxiv.org/abs/2306.03341.

[14] Thomas McGrath, Daniel Balsam, Myra Deng, and Eric Ho. Understanding and steering llama 3 with sparse autoencoders. Goodfire Research, September 2024. URL https://www.goodfire.ai/papers/understanding-and-steering-llama-3. Online.

[15] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt, 2023. URL https://arxiv.org/abs/2202.05262.

[16] Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling, 2025. URL https://arxiv.org/abs/2501.19393.

[17] Nina Panickssery, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Matt Turner. Steering llama 2 via contrastive activation addition, 2024. URL https://arxiv.org/abs/2312.06681.

[18] Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Tom Lieberum, Vikrant Varma, János Kramár, Rohin Shah, and Neel Nanda. Improving dictionary learning with gated sparse autoencoders, 2024. URL https://arxiv.org/abs/2404.16014.

[19] Senthooran Rajamanoharan, Tom Lieberum, Nicolas Sonnerat, Arthur Conmy, Vikrant Varma, János Kramár, and Neel Nanda. Jumping ahead: Improving reconstruction fidelity with jumprelu sparse autoencoders, 2024. URL https://arxiv.org/abs/2407.14435.

[20] Mattia Rigotti, Omri Barak, Melissa R Warden, Xiao-Jing Wang, Nathaniel D Daw, Earl K Miller, and Stefano Fusi. The importance of mixed selectivity in complex cognitive tasks. *Nature*, 497(7451):585–590, 2013. doi: 10.1038/nature12160.

[21] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools, 2023. URL https://arxiv.org/abs/2302.04761.

[22] Nishant Subramani, Nivedita Suresh, and Matthew E. Peters. Extracting latent steering vectors from pretrained language models, 2022. URL https://arxiv.org/abs/2205.05124.

[23] Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L Turner, Callum McDougall, Monte MacDiarmid, C. Daniel Freeman, Theodore R. Sumers, Edward Rees, Joshua Batson, Adam Jermyn, Shan Carter, Chris Olah, and Tom Henighan. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. *Transformer Circuits Thread*, 2024. URL https://transformer-circuits.pub/2024/scaling-monosemanticity/index.html.

[24] Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J. Vazquez, Ulisse Mini, and Monte MacDiarmid. Steering language models with activation engineering, 2024. URL https://arxiv.org/abs/2308.10248.

[25] Constantin Venhoff, Iván Arcuschin, Philip Torr, Arthur Conmy, and Neel Nanda. Understanding reasoning in thinking language models via steering vectors, 2025. URL https://arxiv.org/abs/2506.18167.

[26] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023. URL https://arxiv.org/abs/2201.11903.

[27] Zhengxuan Wu, Aryaman Arora, Atticus Geiger, Zheng Wang, Jing Huang, Dan Jurafsky, Christopher D. Manning, and Christopher Potts. Axbench: Steering llms? even simple baselines outperform sparse autoencoders, 2025. URL https://arxiv.org/abs/2501.17148.

[28] Zhengxuan Wu, Qinan Yu, Aryaman Arora, Christopher D. Manning, and Christopher Potts. Improved representation steering for language models, 2025. URL https://arxiv.org/abs/2505.20809.

[29] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models, 2023. URL https://arxiv.org/abs/2210.03629.

[30] Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, and Guoyin Wang. Instruction tuning for large language models: A survey, 2025. URL https://arxiv.org/abs/2308.10792.

[31] 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, 2024. URL https://arxiv.org/abs/2309.11998.

A Technical Appendices and Supplementary Material

A.1 Prompts

A.1.1 Concept Selection Prompt

This is a set of example <QuestionType> questions with <N> choices each: <ValidChoices>. Each sentence is provided, followed by the choices, and the answer key. Please consider the questions, and come up with a list of 10 "semantically separate conceptual features" that should be used to answer questions of this type. First, carefully reason about the question types, and then at the end provide the refined list in short descriptive sentences. The conceptual features should be written as instructions, as if informing an assistant on the optimal ways to think or respond. Example 1,..., Example 24

A.1.2 Question Answering Prompt

Xwinograd

You are a helpful assistant. You will be presented with a fill in the blank sentence with two possible choices. Briefly think about the question, then provide the answer at the end.

XNLI

You are a helpful assistant. You will be presented with a premise and a hypothesis with three possible choices for the logical relationship between the premise and hypothesis. Briefly think about the question, then provide the answer at the end.

Amazon Reviews

You are a helpful assistant. You will be presented with a review of a product with the task of classifying the sentiment in the review as either positive or negative. Briefly think about the review, then provide the answer at the end."

A.1.3 Formatter Prompt

You are a helpful assistant that extracts the multiple choice answer from another assistant's response. The other assistant will respond with the semantic answer, you will map it to the corresponding letter from the choices. Valid choices are: [A, B,...,None]. Return just the capital letter. If the previous assistant did not provide a valid, extractable answer or provided a response not within the choices, return None. If it appears that the previous assistant made a mistake, do not correct it. The choices provided to the previous assistant are: Choices. The previous assistant's response is: "Response."

A.2 LLM-Selected Features

LLM-Selected SAE Features for XWinograd

```
"Introduction of multiple causal factors in analytical explanations"
"Instructions to identify causal relationships between events"
"connecting actions with their intended purposes or outcomes"
"Groups of characters described in collective states or actions"
"Analytical explanations of emotions and desires"
"Speculative discussion of emotional states and feelings"
"Temporal and sequential ordering concepts"
```

```
"Temporal and sequential transition phrases"
"Material properties and characteristics (physical and abstract)"
"Scientific observation and analysis verbs"
"Political and social power dynamics"
"Analysis of power dynamics and systemic relationships"
"Ownership and possession concepts across domains"
"Institutional representation and possession relationships"
"Discussion of domain-specific knowledge or expertise"
"Academic technical terminology in educational contexts"
"Basic syntactic glue and connecting elements in English and code"
"Complex explanatory sentence structures in formal writing"
"The model needs to verify factual consistency"
"Evaluating factual consistency or correctness"
```

LLM-Selected SAE Features for XNLI

```
"Cross-lingual detection of semantic difference/distinctness"
"Assessment of potential, capability, or relationship between things"
"Structural elements and vocabulary characteristic of different writing styles
    and genres"
"Contradiction or correction of previous statements"
"Technical boilerplate code for imports and API initialization"
"Technical explanation patterns showing causation or implication"
"Verification that facts are explicitly mentioned rather than implied or
    hallucinated"
"Logical inference and deduction processes"
"Inclusive/comprehensive scope markers in formal language"
"Articles and quantifiers across multiple languages"
"Temporal and conditional relationships marked by 'when'"
"Temporal relationships in explanatory contexts"
"Detection of content characterized as having deeper significance or meaning"
"Syntactic markers that precede new information or content"
"Measuring or analyzing sentiment and feedback"
"Formal positive sentiment analysis and evaluation"
"Fact-checking and verification of factual accuracy"
"The model should emphasize factual accuracy and resist making things up"
"Formal logical reasoning and argumentation"
"Start of conditional statements requiring logical analysis or ethical
    consideration"
```

LLM-Selected SAE Features for Amazon Reviews

```
"Comparing subjects to their own expectations or requirements"
"Comparing actual outcomes to expectations, especially when they differ"
"Measuring or evaluating performance and effectiveness"
"Analytical statements about performance or capability"
"Measuring and evaluating satisfaction levels in formal contexts"
"Gradation words in formal measurement scales"
"Software bugs and defect tracking"
"The assistant is listing drawbacks or limitations in a numbered format"
"Explaining how something is perceived or interpreted by others"
"Active perception or consideration from a specific viewpoint"
"Emphatic denial or strong refutation"
"The assistant is evaluating or analyzing the quality of responses"
"Technical implementation and setup instructions"
"User interface design qualities emphasizing ease of use"
"Product descriptions emphasizing durability and quality construction"
"Discussion of final product quality in manufacturing contexts"
"Technical discussion of recommendation systems"
"Technical explanations of recommendation systems and algorithms"
"Technical problems or issues requiring troubleshooting"
"Resolution or handling of problems, errors, or conflicts"
```

A.3 Random Features

Randomly Selected SAE Features

```
"Spanish and Indonesian grammatical connectors and elaborative suffixes"
"The assistant lacks knowledge about a specific entity"
"Bankruptcy and insolvency proceedings across languages"
"Tokens marking stages in data or reasoning transformations"
"The assistant should begin generating creative content based on specifications"
"The assistant is transitioning to provide a detailed explanation or list"
"The letter combination 'sk', particularly in Swedish words and technical terms"
"The assistant is providing instructional examples or explaining possibilities
    to the user"
"Describing lasting influence and legacy in biographical writing"
"Descriptions of terrorist attacks and military strikes"
"Introductory descriptions of prestigious international events and competitions"
"Discussions requiring careful handling of race-related topics"
"Technical terms beginning with pre-"
"Explanations of function return behavior in programming documentation"
"Professional job positions and roles in formal contexts"
"Questions probing for characteristics in world-building and hypothetical
    scenarios"
"Modal and auxiliary verbs expressing possibility or uncertainty"
"Descriptions of mystical energy being harnessed or transferred"
"The assistant is using explicit conditional statements to enumerate different
    cases and their implications"
"Comparative frequency and recency of events or behaviors"
```

A.4 Baselines

Table 1: Dataset baselines

	XWinograd			**XNLI**			**Amazon**		
	Train	Val	Test	Train	Val	Test	Train	Val	Test
Llama-3.1-8B-Instruct	0.706	0.682	0.682	0.71	0.654	0.675	0.936	0.926	0.941
Llama-3.1-8B-Instruct w/ SAE	0.622	0.7	0.701	0.678	0.619	0.672	0.931	0.929	0.945
GPT-4.1-mini	0.926	0.872	0.893	0.866	0.856	0.876	0.956	0.952	0.951

A.5 Example Best Feature Weights for Xwinograd English

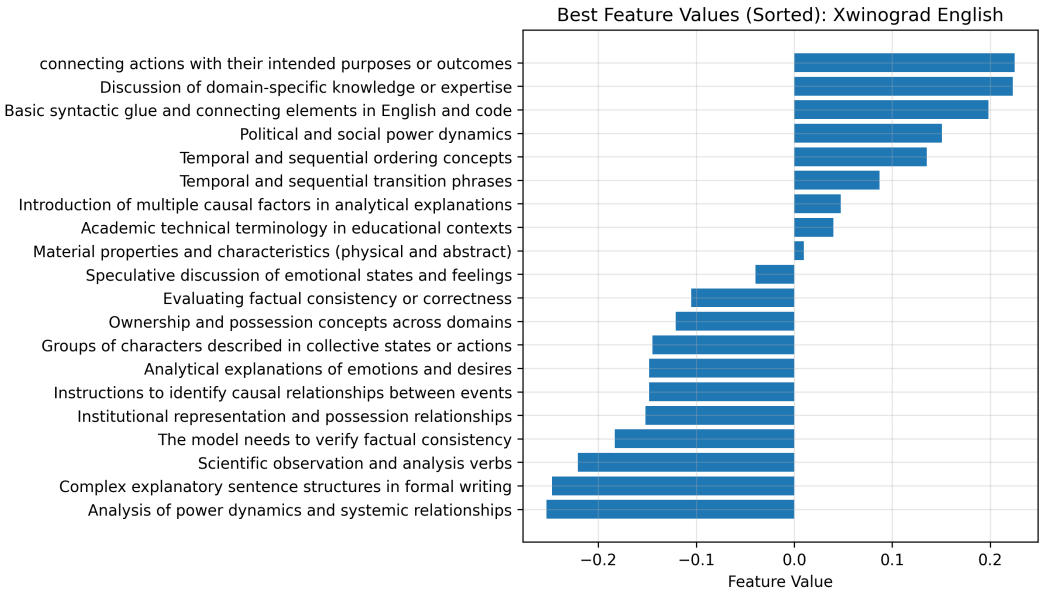

Figure 4: Best performing semantically selected steering feature weights on the Xwinograd English task.