Toward Efficient Sparse Autoencoder-Guided Steering for Improved In-Context Learning in Large Language Models

Ikhyun Cho and **Julia Hockenmaier**
University of Illinois at Urbana-Champaign
{ihcho2, juliahmr}@illinois.edu

Abstract

Sparse autoencoders (SAEs) have emerged as a powerful analytical tool in mechanistic interpretability for large language models (LLMs), with growing success in applications beyond interpretability. Building on this momentum, we present a novel approach that leverages SAEs to enhance the general in-context learning (ICL) performance of LLMs.

Specifically, we introduce **F**eature **D**etection through **P**rompt **V**ariation (FDPV), which leverages the SAE's remarkable ability to capture subtle differences between prompts, enabling efficient feature selection for downstream steering. In addition, we propose a novel steering method tailored to ICL—**S**elective **I**n-Context **Ste**e**r**ing (SISTER)—grounded in recent insights from ICL research that LLMs utilize label words as key anchors. Our method yields a 3.5% average performance improvement across diverse text classification tasks and exhibits greater robustness to hyperparameter variations compared to standard steering approaches. Our code is available at https://github.com/ihcho2/SAE-ICL.

1 Introduction

Sparse autoencoders (SAEs) have recently emerged as a powerful tool in the field of mechanistic interpretability (MI), which seeks to understand and explain how large language models (LLMs) generate their outputs (Cunningham et al., 2023; Sharkey et al., 2025). Trained in an unsupervised manner with a sparsity constraint, SAEs have been shown—albeit to some extent—to effectively decompose LLM embeddings into sparse features that align with human-interpretable, mono-semantic concepts (Sharkey and Beren, 2022; Bricken et al., 2023). Encouraged by their potential, researchers have begun exploring SAE applications beyond interpretability, including probing (Kantamneni et al., 2025), analyzing SAE features tied to specific domains like reinforcement learning (Demircan et al., 2024), and steering LLM outputs toward desired behaviors using SAE features (Bricken et al., 2023; Wu et al., 2025). Building on this momentum, we show that SAEs can also be effectively harnessed to improve in-context learning (ICL) performance—a core capability of modern LLMs.[1]

A central challenge in leveraging SAEs for ICL is determining which features to use for steering, given the sheer number of features—ranging from 16,000 to millions (e.g., GemmaScope) (Lieberum et al., 2024). Existing feature analysis techniques—such as computing indirect effects or performing attribution patching by ablating individual or groups of features (Kharlapenko et al., 2025; Jing et al., 2025)—are computationally intensive. They require multiple model runs under different ablation combinations, which becomes prohibitively expensive particularly for SAEs with such a large number of features. Moreover, these methods have been validated primarily on relatively simple tasks like indirect object identification (Kissane et al., 2024) or subject–verb agreement (Marks et al., 2024), limiting their generalizability. Recent work has also raised concerns about the robustness of commonly used circuit metrics—such as faithfulness (Miller et al., 2024; Kharlapenko et al., 2025)—further underscoring the need for more efficient and reliable approaches. Additionally, even after identifying relevant features, the optimal strategy for intervening on them remains an open question.

These challenges motivate the following research questions, which we aim to address in this paper:

RQ1: How can we efficiently identify effective SAE features for steering in a given task?

[1] See Section 2.1 for an overview of in-context learning.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 28950–28962
November 4-9, 2025

RQ2: Given the identified features, what is an effective and robust method for steering them?

In this paper, we propose **F**eature **D**etection through **P**rompt **V**ariation (FDPV)—a fully unsupervised and computationally efficient method for identifying SAE features suitable for steering. The central idea is to leverage SAEs' ability to detect subtle differences between prompt variations, in contrast to prior work that primarily focused on feature ablation with a fixed prompt. Our key intuition is that when one prompt (i.e., Prompt-Variant) consistently outperforms the baseline prompt (i.e., Prompt-Original) on a given task, comparing their SAE activation patterns can reveal meaningful insights. In particular, features that are consistently more (or less) active in Prompt-Variant are likely to contribute positively (or negatively) to task performance and are therefore strong candidates for steering.

We leverage a recent finding in prompt engineering—namely RE2 (Xu et al., 2024), that demonstrates simply repeating the test query can consistently improve performance across a wide range of tasks— as a representative example as the Prompt-Variant to showcase the effectiveness of FDPV.

Regarding the second research question, we draw inspiration from recent advances in ICL research, which suggest that LLMs use label words as anchors to perform ICL classification (Wang et al., 2023; Cho et al., 2024). Building on this insight, we propose **S**elective **I**n-Context **Steer**ing (SISTER), a method that focuses intervention primarily on label words, rather than across all tokens as in standard approaches. This targeted strategy not only shows direct and pronounced effect on performance but also reduces the number of tokens being manipulated, resulting in greater robustness to hyperparameters compared to conventional steering methods.

Our main contributions are as follows:

1. We propose an unsupervised and computationally efficient method for identifying candidate SAE features for steering which is applicable to general ICL classification tasks.

2. We introduce a novel steering method tailored to ICL—*Selective In-Context Steering*—grounded in recent findings in ICL research (Wang et al., 2023).

3. While recent studies have begun exploring the use of SAEs in ICL, to the best of our knowledge, this work is among the first—if not the very first—to demonstrate direct improvements in general ICL task performance.

2 Related Work

2.1 In-Context Learning

With the rapid advancement of large language models (LLMs), In-Context Learning (ICL) has emerged as a key capability—enabling LLMs to perform various tasks by inferring patterns on-the-fly from just a few examples presented within a single prompt. Because it requires no additional parameter updates, ICL offers a highly efficient alternative to traditional fine-tuning, making it a dominant paradigm in natural language processing (Brown et al., 2020). In ICL, the model's prediction for a test instance is conditioned on the task instruction and the provided n-shot exemplars. For a more detailed overview, we refer readers to (Dong et al., 2022).

Recent findings in ICL classification tasks suggest that LLMs generally use the label words from few-shot exemplars as anchors—forming representations of the label space in the earlier layers and then leveraging them in the upper layers for prediction (Wang et al., 2023). This implies that the hidden states of label words play a particularly important role in ICL performance. Motivated by this insight, we propose SISTER, which focuses steering specifically on these anchor tokens to achieve a more targeted and effective performance boost. An example of label words in ICL classification prompts is provided in Figure 5 in Appendix A.

2.2 Sparse Autoencoders

Understanding the precise internal workings of LLMs has been a longstanding goal in mechanistic interpretability (MI). Recent studies in MI suggest the *linear representation hypothesis*, which posits that semantic concepts are often encoded as linear directions in the hidden representation space of LLMs (Jiang et al., 2024; Li et al., 2023). Building on this hypothesis, along with the *superposition hypothesis*—which posits that LLMs encode more concepts than the representation dimension they have, causing multiple concepts to be entangled within single neurons—researchers have developed Sparse Autoencoders (SAEs). These models learn directions utilizing a much larger, over-complete basis space and have been shown to effectively and sparsely disentangle semantic features from LLM

hidden states (Sharkey and Beren, 2022; Templeton, 2024).

A standard SAE consists of an encoder and a decoder, where each column of the decoder weight matrix ($W_{dec} \in \mathbb{R}^{d \times s}$) represents an SAE feature vector in $\mathbb{R}^d$, with d denoting the dimensionality of the LLM's hidden state. Given a hidden state, the SAE encoder learns to produce a sparse activation over these features. The degree of sparsity depends on the hyperparameters, but the SAEs we use—referred to as the "canonical" version in Gemma Scope—typically activate around a few hundred features on average.

General SAEs are trained using a combination of a reconstruction loss (squared error) and a sparsity regularization term:

$$L_{SAE} = \frac{1}{N} \sum_{i=1}^{N} \|\mathbf{x}^i - \hat{\mathbf{x}}^i\|_2^2 + \lambda \cdot L_{sparsity}$$

where $\mathbf{x}^i$ represents the original input, and $\hat{\mathbf{x}}^i$ is the reconstructed output of the SAE:

$$\hat{\mathbf{x}}^i = W_{dec}\big(\sigma(W_{enc}(\mathbf{x}^i))\big)$$

Various activation functions have been used for σ, including JumpReLU (e.g., Gemma Scope (Lieberum et al., 2024)) and TopK-ReLU (e.g., GPT-4 SAE (Gao et al., 2024) and Llama Scope (He et al., 2024)).

2.3 Sparse Autoencoders for In-Context Learning

Recently, there has been a growing body of work investigating the use of SAEs in the context of ICL. Jing et al. (2025) demonstrate that certain SAE features capture linguistic properties such as phonetics, phonology, and morphology, and further establish causal relationships by intervening on these features. Demircan et al. (2024) show that some SAE features closely align with temporal difference (TD) errors—a core concept in reinforcement learning—and demonstrate that these features play a key role in computing Q-values through targeted interventions. Kharlapenko et al. (2025) attempt to decompose task vectors using SAEs and identify two distinct classes of features: task execution features, which activate when encountering the task, and task completion features, which activate specifically when the task is completed within the prompt.

While prior work has primarily focused on analyzing the properties of SAE features (Cho and Hockenmaier), we switch gears and adopt a more practical perspective—aiming to directly improve general ICL task performance. To the best of our knowledge, our work is the first to systematically demonstrate how SAEs can be harnessed to boost overall ICL effectiveness.

2.4 Standard Steering Approaches

As discussed in Section 2.3, SAEs in the context of ICL are naturally associated with representation-level interventions, commonly referred to as *steering*. Standard steering approaches construct contrastive pairs consisting of positive and negative examples, and derive steering vectors as their difference (Panickssery et al., 2023; Li et al., 2023; Wang et al., 2024). More recently, SAE features have emerged as a promising candidate for steering, though results have been mixed (Kantamneni et al., 2025; Wu et al., 2025).

Nevertheless, most existing steering methods adhere to a relatively standard paradigm—either by adding fixed vectors, such as the sum of selected SAE features, or by clamping specific feature activations to fixed values across all tokens (Templeton, 2024; Wu et al., 2025). We argue that a steering method specifically tailored for ICL-classification can lead to various advantages.

Overall, the main contribution of this work is demonstrating that SAE-based steering can be pushed further through a more efficient feature selection technique (FDPV) and a targeted and effective steering method (SISTER).

3 Enhancing In-Context Learning with Sparse Autoencoders

Overview In this paper, we demonstrate that SAEs can be leveraged to enhance general ICL performance, by effectively addressing the two research questions outlined in the Introduction (RQ1 and RQ2). We introduce **F**eature **D**etection through **P**rompt **V**ariation (FDPV) to address RQ1, and **S**elective **I**n-context **Stee**ring (SISTER) to address RQ2.

3.1 Feature Detection through Prompt Variations (FDPV)

Core Intuition We propose **F**eature **D**etection through **P**rompt **V**ariation (FDPV), a simple yet effective method for identifying candidate SAE features for steering. The central idea is that *"If a prompt (i.e., prompt-variant) outperforms a baseline prompt (i.e., prompt-original), then features*

that are significantly more activated in the better-performing prompt are likely beneficial for the task—making them strong candidates for steering." As shown in Section 4.3, thanks to the strong capability of SAEs to detect subtle differences between prompts, FDPV can easily identify effective SAE features for steering in a fully unsupervised and efficient manner—without relying on any costly metrics.

Methodology Based on this insight, FDPV operates as follows: Given a development set (D_{dev}) along with a Prompt-Original and a Prompt-Variant—where, without loss of generality, the variant performs better than the original on the dev set—we run the LLM on all examples in the dev set using both prompts. For each input, we extract the hidden state of **the last token** from a specific layer (e.g., a middle layer) and apply the SAE encoder to obtain the corresponding feature activations:

$$\begin{cases} z^l_{orig} = \mathcal{S}^l_{enc}(h^l_{orig}) \\ z^l_{var} = \mathcal{S}^l_{enc}(h^l_{var}) \end{cases}, h^l \in R^d \rightarrow z^l \in R^s$$

Here, h^l denotes the hidden state of the last token at layer l, $\mathcal{S}_{enc}$ refers to the SAE encoder, and z^l is the resulting activation vector. The dimensions d and s correspond to the size of the hidden state and the width of the SAE encoder, respectively. The motivation for using only the last token is that if an SAE feature genuinely contributes to improved performance, it should be salient enough to appear at the last token—where the model ultimately makes its prediction. Focusing on the last token also helps narrow down the candidate feature set from the start.

Then, for each SAE feature index $\forall i \in \{1, 2, \ldots, s\}$, we first discard features that are activated in fewer than 30% of the development examples. This follows the same intuition that truly task-relevant features should appear with reasonable frequency at the last token across the dev set.

Next, we compute two scores—f^+_{FDPV} and f^-_{FDPV}—which quantify how often a feature's activation at the last token is stronger or weaker, respectively, in the Prompt-Variant:

$$f^+_{\text{FDPV}}(i) = \frac{\sum_{j=1}^{|D_{dev}|} \mathbb{1}\left[z^l_{var,j}(i) > z^l_{orig,j}(i)\right]}{|D_{dev}|}$$

$$f^-_{\text{FDPV}}(i) = \frac{\sum_{j=1}^{|D_{dev}|} \mathbb{1}\left[z^l_{var,j}(i) < z^l_{orig,j}(i)\right]}{|D_{dev}|}$$

where $\mathbb{1}$ represents the indicator function. Note that in general $f^+_{\text{FDPV}}(i) \neq 1 - f^-_{\text{FDPV}}(i)$ because SAE features often remain inactive (i.e., $z^l_{var,j}(i) = z^l_{orig,j}(i) = 0$) due to their inherently sparse nature.

A high f^+_{FDPV} (or f^-_{FDPV}) value indicates features that are consistently more (or less) activated in the Prompt-Variant compared to the Prompt-Original, suggesting a potentially positive (or negative) impact on task performance and making them strong candidates for steering. We then select the top-K features (with K chosen from $\{0, 1, 3, 5\}$) with the highest f^+_{FDPV} and f^-_{FDPV} scores, respectively:

$$\text{LIST}^+_{\text{FDPV}} = \text{TopK}(f^+_{\text{FDPV}}(i), \forall i)$$
$$\text{LIST}^-_{\text{FDPV}} = \text{TopK}(f^-_{\text{FDPV}}(i), \forall i)$$

Finally, we define the steering vector used for intervention as follows:

$$\mathcal{V}_{\text{FDPV}} = \sum_{i \in \text{LIST}^+_{\text{FDPV}}} W_{dec}(i) - \sum_{i \in \text{LIST}^-_{\text{FDPV}}} W_{dec}(i)$$

Where $W_{\text{dec}}(i)$ denotes the i-th column of the SAE decoder weight, corresponding to the i-th feature.

FDPV enables a fully unsupervised and computationally efficient search for SAE features, requiring only a single forward pass per prompt and item. Figure 1 presents a visual overview of FDPV and deeper insights into FDPV are discussed in Section 4.3.

3.2 Selective In-context Steering (SISTER)

Core Intuition Since our focus is on ICL classification tasks, we hypothesize that a steering method specifically tailored to this setting can yield superior performance. To this end, we propose a novel steering approach—**S**elective **I**n-context **Ste**e**r**ing (SISTER)—explicitly designed for ICL classification and grounded in recent insights from ICL research. Motivated by findings that LLMs use label words from few-shot exemplars as anchors (Wang et al., 2023)—*implying that the hidden states of these tokens play a central role*—we propose steering these specific tokens to exert a more direct and pronounced influence on performance.

Specifically, instead of applying the steering vector to all tokens—as done in standard steering (Bricken et al., 2023)—we focus on a smaller, more stable subset. We demonstrate that label words are particularly well-suited for this purpose, achieving improved performance over the standard approach.

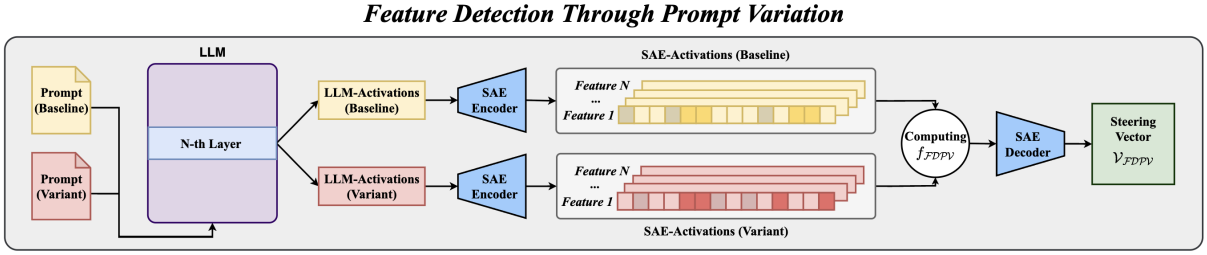


Figure 1: **Overall Architecture of Feature Detection through Prompt Variation (FDPV).** FDPV is built on SAEs' surprising ability to capture subtle distinctions between different prompts and requires only a single forward pass per prompt. FDPV generates a steering vector $\mathcal{V}_{\text{FDPV}}$,which is subsequently used for steering.

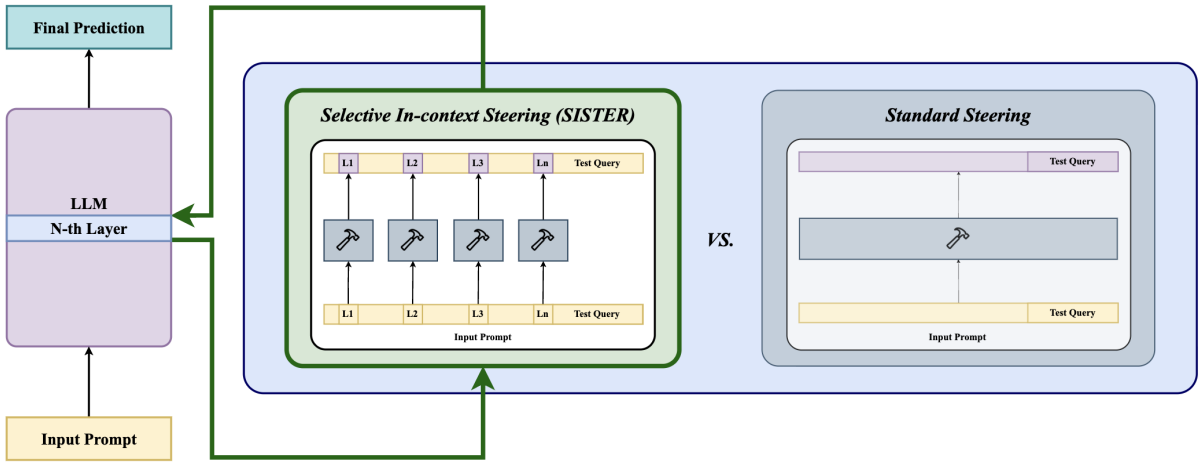


Figure 2: **Overall Architecture of Selective In-context Steering (SISTER).** Using $\mathcal{V}_{\text{FDPV}}$, SISTER intervenes only on key anchor tokens, yielding more targeted and effective results. Moreover, by operating on far fewer tokens than standard steering—leaving the test query entirely untouched—it achieves greater stability with respect to the hyperparamter α.

Furthermore, by restricting intervention to a limited set of tokens (i.e., label tokens) while leaving the test query untouched, our method exhibits increased robustness to variations in the hyperparameter α, as discussed in Section 4.4.

Methodology Given an n-shot ICL prompt with a steering vector $\mathcal{V}_{\text{FDPV}}$ derived from FDPV, SISTER operates as follows:

$$\forall i \in \{L_1, L_2, \cdots, L_n\}: \quad (1)$$

$$\bar{h}_i^l = h_i^l + \alpha \cdot A_{\max} \cdot \mathcal{V}_{\text{FDPV}} \quad (2)$$

Where $L_1, L_2, \cdots, L_n$ indicate the label words, $\bar{h}$ represents the steered representation, $A_{\max}$ denotes the maximum activation value of the features in $\text{LIST}^+_{\text{FDPV}}$ and $\text{LIST}^-_{\text{FDPV}}$ across the development set, and the hyperparameter α controls the strength of the intervention. Unlike standard steering methods that intervene on all tokens, SISTER applies interventions only to the label words ($n \ll N$), resulting in reduced sensitivity to the choice of α. Moreover, rather than clamping activations to fixed values as in standard steering, it allows the activations in the test query to be adjusted more naturally by the LLM, potentially leading to more flexible and effective outcomes. Figure 2 provides a visual overview of SISTER.

4 Experiments and Analyses

4.1 Experimental Settings

Datasets We evaluated our approach on four widely used ICL classification datasets. Specifically, we used aspect-based sentiment classification (ABSC) with SemEval-14 Laptops and Restaurants (Pontiki et al., 2014), news topic classification with AGNews (Zhang et al., 2015), and emotion classification from short dialogues using EmoC (Chatterjee et al., 2019).

Models and Settings Our approach requires access to well-trained SAEs for the target LLMs. Given the high computational cost of training SAEs

Model	Tasks				
1. Gemma2-9B-IT	AGNews	Rest14	Lap14	EmoC	Avg.
ICL-Baseline	$84.08_{1.73}$	$79.67_{1.65}$	$73.71_{1.12}$	$68.70_{2.38}$	$76.54_{1.87}$
RE2-style	$84.62_{1.88}$	$80.10_{1.79}$	$75.01_{1.45}$	$70.00_{2.52}$	$77.43_{1.91}$
FDPV + Standard Steering	$84.88_{1.79}$	$81.12_{1.33}$	$76.29_{1.50}$	$69.88_{1.96}$	$78.04_{1.64}$
(Ours) FDPV + SISTER	$\mathbf{86.26_{1.80}}$	$\mathbf{82.76_{0.91}}$	$\mathbf{78.97_{1.02}}$	$\mathbf{71.99_{1.04}}$	$\mathbf{80.00_{1.29}}$
2. Gemma2-2B-IT	AGNews	Rest14	Lap14	EmoC	Avg.
ICL-Baseline	$70.39_{5.69}$	$74.60_{1.43}$	$72.22_{1.74}$	$60.74_{7.86}$	$69.49_{4.18}$
RE2-style	$71.93_{3.97}$	$76.03_{1.63}$	$73.90_{1.24}$	$61.15_{5.96}$	$70.75_{3.20}$
FDPV + Standard Steering	$75.12_{3.84}$	$75.99_{1.95}$	$75.20_{1.88}$	$61.85_{5.72}$	$72.04_{3.35}$
(Ours) FDPV + SISTER	$\mathbf{77.39_{2.34}}$	$\mathbf{77.22_{2.10}}$	$\mathbf{76.08_{1.52}}$	$\mathbf{62.63_{6.01}}$	$\mathbf{73.33_{2.99}}$
3. Llama3-8B-IT	AGNews	Rest14	Lap14	EmoC	Avg.
ICL-Baseline	$79.38_{2.51}$	$76.13_{1.87}$	$74.79_{1.52}$	$65.62_{3.55}$	$73.98_{2.36}$
RE2-style	$80.62_{2.18}$	$76.99_{1.76}$	$75.55_{1.62}$	$66.61_{3.88}$	$74.94_{2.36}$
FDPV + Standard Steering	$81.55_{2.30}$	$77.37_{1.75}$	$76.02_{1.44}$	$67.09_{2.86}$	$75.51_{2.09}$
(Ours) FDPV + SISTER	$\mathbf{83.91_{1.26}}$	$\mathbf{79.02_{0.81}}$	$\mathbf{77.50_{1.04}}$	$\mathbf{68.33_{2.67}}$	$\mathbf{77.19_{1.44}}$

Table 1: **Effectiveness of SISTER.** We observe that SISTER consistently outperforms the standard steering approach commonly adopted in recent studies, supporting our hypothesis that steering methods specifically tailored for ICL can lead to substantial performance gains. The reported results are averaged over 15 random seeds, as described in Section 4.1, using the mean of accuracy and F1 score as the evaluation metric.

from scratch, we leverage publicly available SAEs, Gemma Scope (Lieberum et al., 2024) and Llama Scope (He et al., 2024). Accordingly, we conduct experiments using three main models—Gemma2-9B-IT, Gemma2-2B-IT, and Llama3-8B-IT—along with their corresponding SAEs. We use SAEs from either the middle layer or from a layer located at approximately five-sixths of the model depth, following common practice (Gao et al., 2024).

We follow the standard ICL prompt format, which includes a task instruction, n-shot exemplars, and a test query. For each task, few-shot exemplars are randomly sampled from the training data using five different random seeds. Additionally, the order of the exemplars is randomly shuffled three times per seed, resulting in a total of 15 distinct prompts evaluated per task. This setup accounts for the well-established finding that both the choice and order of exemplars can significantly influence ICL performance (Guo et al., 2024; Ye et al., 2023). We use the average of accuracy and f1 score as our primary metric. For the development set used by FDPV, we randomly sampled 100 examples per label from the training data. Full experimental details are provided in Appendix A.

4.2 Overall Results

Table 1 presents the overall experimental results. Since there are no established steering variants beyond the standard approach that leverage SAEs for improving general ICL classification, we directly compare SISTER with the standard steering baseline. This baseline uses the same steering vector as SISTER but applies it indiscriminately across all tokens, following prior work (Liu et al., 2023; Templeton, 2024).

Table 1 highlights two key findings: (1) Standard steering also benefits from using the steering vector generated by FDPV (i.e.,$\mathcal{V}_{\text{FDPV}}$), indicating that FDPV effectively identifies features suitable for steering; and (2) SISTER consistently outperforms the standard steering baseline across all tasks and models, underscoring its broad effectiveness. Moreover, we show in Section 4.4 that SISTER is markedly more robust to changes in the hyperparameter α, providing yet another advantage over conventional methods.

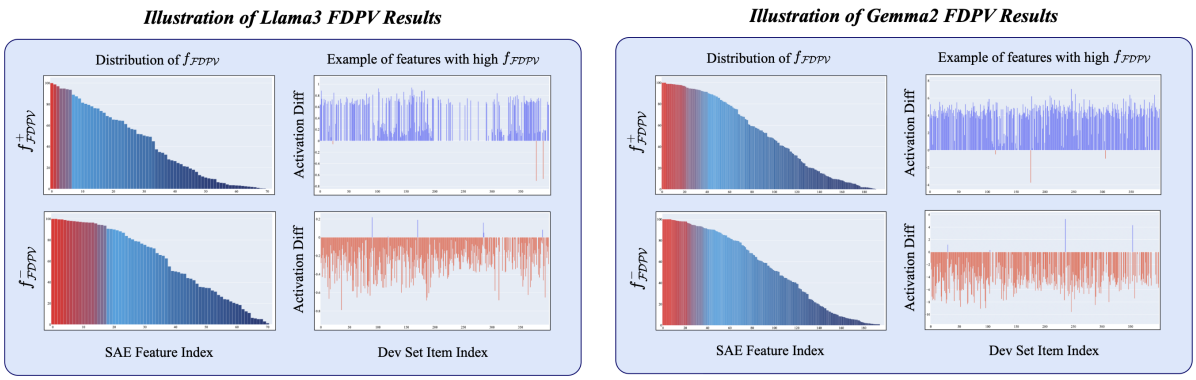


Figure 3: **SAEs Exhibit a Strong Capacity to Capture Subtle Prompt Distinctions.** We demonstrate this through two case studies—AGNews with Gemma2-9B-IT and AGNews with Llama3-8B-IT. The first columns show the overall distributions of f^+_{FDPV} and f^-_{FDPV}, while the second columns show examples of highly skewed features that clearly distinguish the two prompts. Two key findings emerge: (1) SAEs exhibit remarkable sensitivity, with multiple features reaching near 1.0 scores on either metric, demonstrating their ability to detect even subtle prompt differences; and (2) this behavior appears consistently across both Llama3 and Gemma2, suggesting it is a general property of SAEs.

4.3 Foundation of FDPV: SAEs' Remarkable Capability to Detect Subtle Differences between Prompts

FDPV builds on the ability of SAEs to detect subtle distinctions between the Prompt-Original and Prompt-Variant. In other words, if SAEs were unable to capture these nuanced variations, FDPV would not be effective. However, as Figure 3 shows, SAEs are remarkably sensitive to these variations. Specifically, we identify several features (highlighted in red) that are consistently and substantially more activated in the Prompt-Variant than in the Prompt-Original—some even approaching a 100% f^+_{FDPV} score. Conversely, we also find features that exhibit the opposite behavior, with consistently lower activation in the Prompt-Variant—several nearing a 100% f^-_{FDPV} score.

This pronounced skew is particularly surprising for two key reasons. First, although the Prompt-Variant simply repeats the test query without adding any new semantic content, SAEs still register strong and consistent activation differences—suggesting that even slight shifts in hidden states caused by prompt variations are effectively picked up by SAE features. Second, because SAEs are trained in a bag-of-embeddings fashion without any prompt-level supervision, there is no inherent guarantee they should consistently produce stronger (or weaker) activations for one semantically equivalent prompt over another. These factors make the observed behavior especially striking. Notably, both Gemma2-9B-IT and Llama3-8B-IT show highly similar activation patterns, indicating that this strong sensitivity is likely a general strength of SAEs.

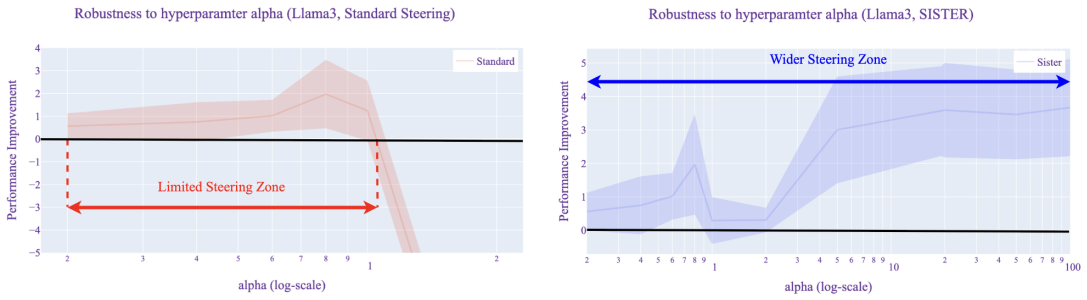
To the best of our knowledge, our work is the first to leverage this unique sensitivity of SAEs, and we believe it opens up exciting directions for future research. For instance, SAEs could potentially capture nuanced differences in multilingual parallel corpora—where the same semantics are expressed in different languages—offering deeper insights into how LLMs represent and process language across linguistic boundaries.

Overall, this sensitivity of SAEs forms the foundation of FDPV, enabling it to operate more efficiently and reliably than existing approaches that require multiple repeated model runs and rely on external evaluation metrics.

4.4 SISTER Exhibits a Broader Goldilocks Zone

One limitation of standard steering approaches is that they often manipulate most, if not all, tokens in the input prompt, likely due to the lack of a systematic method for selecting which tokens to steer. Intervening on many tokens naturally leads to a strong sensitivity to the magnitude hyperparameter α, which is undesirable. In contrast, SISTER sidesteps this issue by operating on a much smaller, more targeted set of tokens, resulting in significantly greater robustness to α compared to standard steering methods.

Figure 4 shows test performance improvements



(a) Limited Goldilocks Zone of Standard Steering

(a) Much Wider Goldilocks Zone of SISTER

Figure 4: SISTER exhibits a much wider Goldilocks zone compared to the standard steering approach. Experiments conducted on Lap14 and AGNews with Llama3-8B-IT. Shaded area represents standard deviation.

over the baseline (i.e., without steering) across varying values of the hyperparameter α, comparing SISTER with the standard approach on AGNews and Lap14 using Llama3-8B-IT. As illustrated, SISTER achieves a significantly broader Goldilocks zone—that is, a range of α values for which performance consistently exceeds the baseline.

As shown in Figure 4 (a), the standard steering method is effective only within a narrow range of α, roughly up to 1. In contrast, SISTER maintains improved performance across an exceptionally wide range of α values—essentially an unbounded Goldilocks zone—with consistently higher gains. Notably, given that the x-axis is in log scale, the breadth of this stable region is both substantial and surprising. Even at large α values, performance remains stable without degradation. We speculate that this robustness arises because SISTER intervenes only on the label words, allowing the model to retain flexibility in processing the rest of the test query. This selective intervention likely prevents the kind of performance collapse observed in standard steering, which modifies all token representations indiscriminately.

4.5 Orthogonality of SISTER to Prompt Selection Methods

Another notable strength of SISTER is its broad effectiveness across prompts with varying exemplars and ordering. Specifically, it improves performance not only on low-performing prompts—those that yield poor baseline results—but also on high-performing ones that already achieve strong results. This behavior suggests that SISTER offers a *complementary and orthogonal* strategy to the widely studied area of prompt selection in ICL, which focuses on optimizing the choice and order of examples in the input prompt.

To examine this in detail, we divided the 15 prompts used for each task into three groups of five based on their oracle baseline test performance: "Great", "Medium", and "Poor." Our goal is for SISTER to improve performance consistently across all three groups. If successful, this would indicate that SISTER can be used alongside existing prompt selection methods to further boost ICL performance. In other words, even if one already has a high-quality prompt through existing prompt selection techniques, SISTER can still "push the boundaries", making it a valuable addition. As shown in Table 2, SISTER improves performance across all groups, confirming its effectiveness regardless of prompt quality.

4.6 Deeper Insights into SISTER

We performed an additional analysis to gain deeper insights into the effectiveness of our approach. Specifically, for each SAE feature index i, we calculated its normalized pointwise mutual information (nPMI) score to assess task specificity. The nPMI is defined as follows: We include prompts from three different tasks (AGNews, ABSC, and ARC). Pointwise mutual information (PMI) between a task X_j and feature f_i is given by:

$$PMI(X_j, f_i) = log\frac{P(X_j, f_i)}{P(X_j)P(f_i)} = log\frac{P(f_i|X_j)}{P(f_i)}$$

$$nPMI(X_j, f_i) = \frac{PMI(X_j, f_i)}{-logP(X_j, f_i)}$$

where, X_j denotes the event that a token belongs to task j, where $j \in$ {AGNews, ABSC, ARC}, and f_i denotes the event that feature i is activated by a token.

We computed nPMI scores for all activated features using Gemma2-2B-IT (Table 4) and observed

1. Gemma2-9B-IT	AGNews Great	AGNews Medium	AGNews Poor	Rest14 Great	Rest14 Medium	Rest14 Poor	Lap14 Great	Lap14 Medium	Lap14 Poor	EmoC Great	EmoC Medium	EmoC Poor
Baseline	$86.76_{0.42}$	$84.07_{0.96}$	$81.91_{1.32}$	$81.52_{0.41}$	$79.47_{0.68}$	$78.02_{1.08}$	$74.92_{0.85}$	$73.66_{0.22}$	$72.56_{0.35}$	$70.63_{1.38}$	$69.03_{0.27}$	$66.45_{2.59}$
Baseline + SISTER	$\mathbf{87.24_{0.46}}$	$\mathbf{86.31_{1.06}}$	$\mathbf{85.24_{1.26}}$	$\mathbf{82.80_{0.83}}$	$\mathbf{82.59_{0.42}}$	$\mathbf{82.89_{1.22}}$	$\mathbf{79.52_{0.88}}$	$\mathbf{79.32_{0.52}}$	$\mathbf{78.07_{1.02}}$	$\mathbf{72.31_{1.04}}$	$\mathbf{72.17_{1.36}}$	$\mathbf{71.50_{0.62}}$
2. Llama3-8B-IT	AGNews Great	AGNews Medium	AGNews Poor	Rest14 Great	Rest14 Medium	Rest14 Poor	Lap14 Great	Lap14 Medium	Lap14 Poor	EmoC Great	EmoC Medium	EmoC Poor
Baseline	$81.73_{1.08}$	$79.97_{0.39}$	$76.44_{1.59}$	$78.34_{1.11}$	$75.72_{0.19}$	$74.34_{0.79}$	$76.10_{0.29}$	$75.28_{0.40}$	$72.99_{1.14}$	$68.85_{0.97}$	$66.05_{1.08}$	$61.96_{3.45}$
Baseline + SISTER	$\mathbf{85.21_{0.68}}$	$\mathbf{83.70_{0.95}}$	$\mathbf{82.80_{0.67}}$	$\mathbf{78.85_{0.98}}$	$\mathbf{79.34_{0.50}}$	$\mathbf{78.87_{0.93}}$	$\mathbf{76.45_{0.86}}$	$\mathbf{78.12_{0.87}}$	$\mathbf{77.93_{0.47}}$	$\mathbf{68.93_{1.71}}$	$\mathbf{68.47_{1.12}}$	$\mathbf{67.60_{3.36}}$

Table 2: **Orthogonality of SISTER to Prompt Selection.** We observe that SISTER is effective throughout all great, medium, and poor quality prompts, indicating the complementariness with existing prompt selection approaches.

that features identified as FDPV + (i.e., features consistently stronger in the better-performing prompt), exhibit higher average nPMI scores than those identified as FDPV- (i.e., consistently weaker features). This suggests that FDPV + features are generally more task-relevant, while FDPV- features are less so. This observation offers insight into our method: SISTER *improves performance by reinforcing task-relevant SAE features while suppressing (relatively) task-irrelevant ones.*

4.7 Beyond Classification to Multiple-Choice Question Answering

Although the key motivation for our SISTER approach arose from observing how LLMs specialize in utilizing label words when solving classification tasks, we extend this idea to multiple-choice question answering (MCQA). Specifically, we evaluate on ARC-Challenge (Clark et al., 2018), using the answer options (A, B, C, and D) as the target tokens for steering. For comparison, we also implement the baseline CAA (Panickssery et al., 2023). Based on the original work, the contrastive pairs consist of a positive example with the correct answer and negative examples with randomly chosen incorrect answers, following recent practice (Wang et al., 2024). The results, summarized in Table 3, show that SISTER is also effective for MCQA.

However, the performance gains are less pronounced than in classification tasks. We speculate that this is because classification tasks make greater use of label words that carry specific semantic meanings, whereas label words in MCQA are simple tokens such as A, B, C, and D.

5 Conclusion

This paper investigates the application of Sparse Autoencoders (SAEs) to In-Context Learning (ICL). While recent studies have begun to analyze the properties of SAE features, to the best of our knowledge, this is the first work to demonstrate their direct impact on general ICL performance and to propose a systematic method for leveraging SAEs in this context. We introduce Feature Detection through Prompt Variation (FDPV), a technique for effectively identifying meaningful features and generating a corresponding steering vector. Building on this, we propose Selective In-Context Steering (SISTER), which applies the steering vector derived from FDPV. Our method achieves substantially greater and more stable performance improvements compared to the standard steering approach.

ARC-Challenge	Gemma2-2B-IT	Gemma2-9B-IT	Llama3-8B-Instruct
ICL Baseline	$74.15_{0.36}$	$90.34_{0.42}$	$68.03_{0.31}$
RE2-Style	$74.38_{0.31}$	$90.44_{0.32}$	$68.20_{0.61}$
CAA	$74.48_{0.45}$	$90.29_{0.49}$	$68.25_{0.53}$
(Ours) SISTER	$\mathbf{75.12_{0.31}}$	$\mathbf{90.90_{0.25}}$	$\mathbf{69.25_{0.47}}$

Table 3: **Effectiveness of SISTER on Multiple-Choice Question Answering Task.** We observe that SISTER is also effective on ARC-Challenge across all models tested, verifying its general effectiveness.

Gemma2-2B-IT	AGNews	ABSC	ARC-Challenge
Avg. nPMI of FDPV + Features	0.5154	0.3116	0.4471
Avg. nPMI of FDPV- Features	0.4436	0.2067	0.4185

Table 4: **Average nPMI Scores of Top FDPV + and FDPV- Featurees.** We observe that FDPV + features are generally more task-relevant compared to FDPV- features.

6 Limitations

A primary constraint of our study is its reliance on sparse autoencoders (SAEs), which are computationally expensive to train and therefore not always readily accessible. As a result, our experiments are based on publicly released SAEs from (Lieberum et al., 2024) and (He et al., 2024), restricting our evaluation to the Gemma2 and Llama3 models. Additionally, due to resource limitations, we were unable to conduct comprehensive layer-wise analyses, which may have yielded further insights. Extending this work to other model architectures and conduct-

ing deeper investigations across layers would be an intriguing direction for future research.

7 Acknowledgment

This research was supported in part by Other Transaction award HR0011249XXX from the U.S. Defense Advanced Research Projects Agency (DARPA) Friction for Accountability in Conversational Transactions (FACT) program. Additionally, this research used the Delta advanced computing and data resource which is supported by the National Science Foundation (award OAC 2005572) and the State of Illinois. Delta is a joint effort of the University of Illinois Urbana-Champaign and its National Center for Supercomputing Applications.

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. 2023. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*. Https://transformer-circuits.pub/2023/monosemantic-features/index.html.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Ankush Chatterjee, Kedhar Nath Narahari, Meghana Joshi, and Puneet Agrawal. 2019. Semeval-2019 task 3: Emocontext contextual emotion detection in text. In *Proceedings of the 13th international workshop on semantic evaluation*, pages 39–48.

Ikhyun Cho and Julia Hockenmaier. Analyzing multilingualism in large language models with sparse autoencoders. In *Second Conference on Language Modeling*.

Ikhyun Cho, Gaeul Kwon, and Julia Hockenmaier. 2024. Tutor-icl: Guiding large language models for improved in-context learning performance. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 9496–9506.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Øyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. 2023. Sparse autoencoders find highly interpretable features in language models. *arXiv preprint arXiv:2309.08600*.

Can Demircan, Tankred Saanum, Akshay K Jagadish, Marcel Binz, and Eric Schulz. 2024. Sparse autoencoders reveal temporal difference learning in large language models. *arXiv preprint arXiv:2410.01280*.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. 2022. A survey on in-context learning. *arXiv preprint arXiv:2301.00234*.

Leo Gao, Tom Dupré la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. 2024. Scaling and evaluating sparse autoencoders. *arXiv preprint arXiv:2406.04093*.

Qi Guo, Leiyu Wang, Yidong Wang, Wei Ye, and Shikun Zhang. 2024. What makes a good order of examples in in-context learning. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 14892–14904.

Zhengfu He, Wentao Shu, Xuyang Ge, Lingjie Chen, Junxuan Wang, Yunhua Zhou, Frances Liu, Qipeng Guo, Xuanjing Huang, Zuxuan Wu, et al. 2024. Llama scope: Extracting millions of features from llama-3.1-8b with sparse autoencoders. *arXiv preprint arXiv:2410.20526*.

Yibo Jiang, Goutham Rajendran, Pradeep Ravikumar, Bryon Aragam, and Victor Veitch. 2024. On the origins of linear representations in large language models. *arXiv preprint arXiv:2403.03867*.

Yi Jing, Zijun Yao, Lingxu Ran, Hongzhu Guo, Xiaozhi Wang, Lei Hou, and Juanzi Li. 2025. Sparse auto-encoder interprets linguistic features in large language models. *arXiv preprint arXiv:2502.20344*.

Subhash Kantamneni, Joshua Engels, Senthooran Rajamanoharan, Max Tegmark, and Neel Nanda. 2025. Are sparse autoencoders useful? a case study in sparse probing. *arXiv preprint arXiv:2502.16681*.

Dmitrii Kharlapenko, Stepan Shabalin, Fazl Barez, Arthur Conmy, and Neel Nanda. 2025. Scaling sparse feature circuit finding for in-context learning. *arXiv preprint arXiv:2504.13756*.

Connor Kissane, Robert Krzyzanowski, Joseph Isaac Bloom, Arthur Conmy, and Neel Nanda. 2024. Interpreting attention layer outputs with sparse autoencoders. *arXiv preprint arXiv:2406.17759*.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023. Inference-time intervention: Eliciting truthful answers from a language model. *Advances in Neural Information Processing Systems*, 36:41451–41530.

Tom Lieberum, Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Nicolas Sonnerat, Vikrant Varma, János Kramár, Anca Dragan, Rohin Shah, and Neel Nanda. 2024. Gemma scope: Open sparse autoencoders everywhere all at once on gemma 2. *arXiv preprint arXiv:2408.05147*.

Sheng Liu, Haotian Ye, Lei Xing, and James Zou. 2023. In-context vectors: Making in context learning more effective and controllable through latent space steering. *arXiv preprint arXiv:2311.06668*.

Samuel Marks, Can Rager, Eric J Michaud, Yonatan Belinkov, David Bau, and Aaron Mueller. 2024. Sparse feature circuits: Discovering and editing interpretable causal graphs in language models. *arXiv preprint arXiv:2403.19647*.

Joseph Miller, Bilal Chughtai, and William Saunders. 2024. Transformer circuit faithfulness metrics are not robust. *arXiv preprint arXiv:2407.08734*.

Nina Panickssery, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Matt Turner. 2023. Steering llama 2 via contrastive activation addition. *arXiv preprint arXiv:2312.06681*.

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. SemEval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 27–35, Dublin, Ireland. Association for Computational Linguistics.

Lee Sharkey and Dan Braun Beren. 2022. [interim research report] taking features out of superposition with sparse autoencoders.

Lee Sharkey, Bilal Chughtai, Joshua Batson, Jack Lindsey, Jeff Wu, Lucius Bushnaq, Nicholas Goldowsky-Dill, Stefan Heimersheim, Alejandro Ortega, Joseph Bloom, Stella Biderman, Adria Garriga-Alonso, Arthur Conmy, Neel Nanda, Jessica Rumbelow, Martin Wattenberg, Nandi Schoots, Joseph Miller, Eric J. Michaud, Stephen Casper, Max Tegmark, William Saunders, David Bau, Eric Todd, Atticus Geiger, Mor Geva, Jesse Hoogland, Daniel Murfet, and Tom McGrath. 2025. Open problems in mechanistic interpretability.

Adly Templeton. 2024. *Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet*. Anthropic.

Lean Wang, Lei Li, Damai Dai, Deli Chen, Hao Zhou, Fandong Meng, Jie Zhou, and Xu Sun. 2023. Label words are anchors: An information flow perspective for understanding in-context learning. *arXiv preprint arXiv:2305.14160*.

Weixuan Wang, Jingyuan Yang, and Wei Peng. 2024. Semantics-adaptive activation intervention for llms via dynamic steering vectors. *arXiv preprint arXiv:2410.12299*.

Zhengxuan Wu, Aryaman Arora, Atticus Geiger, Zheng Wang, Jing Huang, Dan Jurafsky, Christopher D Manning, and Christopher Potts. 2025. Axbench: Steering llms? even simple baselines outperform sparse autoencoders. *arXiv preprint arXiv:2501.17148*.

Xiaohan Xu, Chongyang Tao, Tao Shen, Can Xu, Hongbo Xu, Guodong Long, Jian-Guang Lou, and Shuai Ma. 2024. Re-reading improves reasoning in large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 15549–15575.

Jiacheng Ye, Zhiyong Wu, Jiangtao Feng, Tao Yu, and Lingpeng Kong. 2023. Compositional exemplars for in-context learning. In *International Conference on Machine Learning*, pages 39818–39833. PMLR.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *NIPS*.

A Experimental Details

Detailed statistics of the datasets we used in our study is provided in Table 5.

	Dataset		Label Words	
Task	Train	Test	Label	Count
Lap14	2313	638	Positive	341
			Negative	128
			Neutral	169
Rest14	3602	1120	Positive	728
			Negative	196
			Neutral	196
AGNews	120000	7600	World	1900
			Sports	1900
			Business	1900
			Sci/Fi	1900
EmoContext	30160	5509	Happy	284
			Sad	250
			Angry	298
			Others	4677

Table 5: Detailed statistics of the tasks used in this study.

For the ICL prompts, we follow standard conventions by including a task instruction and k exemplars per label. We set $k = 1$ for AGNews and EmoC, and $k = 2$ for Rest14 and Lap14, as increasing k beyond these values did not yield significant performance gains.

Another key hyperparameter in our approach is α which controls the strength of the steering. We perform a grid search over the values $\{-10, -5, -2, -1, -0.5, 0.5, 1, 2, 5, 10\}$, se-

lecting the value that yields the best performance on the development set.

Label words—referred to as anchors throughout this work—play a central role in our approach, particularly in SISTER, which directly leverages them. An illustrative example of label words is provided in Figure 5 below.

Figure 5: An example of label words in an ICL prompt: each few-shot exemplar includes an answer label, which we refer to as a label word or anchor in this paper (represented in blue).