

---

# Task-Specific Data Selection for Instruction Tuning via Monosemantic Neuronal Activations

---

Da Ma<sup>α</sup>, Gonghu Shang<sup>α</sup>, Zhi Chen<sup>σ</sup>, Libo Qin<sup>δ</sup>, Yijie Luo<sup>α</sup>,  
Hongshen Xu<sup>α</sup>, Lei Pan<sup>γ</sup>, Shuai Fan<sup>γμ</sup>, Kai Yu<sup>αμλ</sup>, Lu Chen<sup>αβμλ\*</sup>,  
<sup>α</sup>X-LANCE Lab, MoE Key Lab of Artificial Intelligence, AI Institute  
School of Computer Science, Shanghai Jiao Tong University, Shanghai, China  
<sup>β</sup>Shanghai Innovation Institution, Shanghai, China  
<sup>γ</sup>AI Speech Co., Ltd., Suzhou, China <sup>σ</sup>ByteDance  
<sup>δ</sup>School of Computer Science and Engineering, Central South University  
<sup>μ</sup>Jiangsu Key Lab of Language Computing, Suzhou, China  
<sup>λ</sup>Suzhou Laboratory, Suzhou, China  
{mada123, chenlusz}@sjtu.edu.cn

## Abstract

Instruction tuning improves the ability of large language models (LLMs) to follow diverse human instructions, but achieving strong performance on specific target tasks remains challenging. A critical bottleneck is selecting the most relevant data to maximize task-specific performance. Existing data selection approaches include unstable influence-based methods and more stable distribution alignment methods, the latter of which critically rely on the underlying sample representation. In practice, most distribution alignment methods, from shallow features (e.g., BM25) to neural embeddings (e.g., BGE, LLM2Vec), may fail to capture how the model internally processes samples. To bridge this gap, we adopt a model-centric strategy in which each sample is represented by its neuronal activation pattern in the model, directly reflecting internal computation. However, directly using raw neuron activations leads to spurious similarity between unrelated samples due to neuron polysemanticity, where a single neuron may respond to multiple, unrelated concepts. To address this, we employ sparse autoencoders to disentangle polysemantic activations into sparse, monosemantic representations, and introduce a dedicated similarity metric for this space to better identify task-relevant data. Comprehensive experiments across multiple instruction datasets, models, tasks, and selection ratios show that our approach consistently outperforms existing data selection baselines in both stability and task-specific performance<sup>2</sup>.

## 1 Introduction

Instruction tuning [1, 2] enables large language models (LLMs) to better follow human instructions, powering versatile applications such as chatbots [3–6]. However, real-world tasks often require specialized abilities—like professional physics problems—that general instruction tuning may not provide [7]. While fine-tuning LLMs on diverse and broad instruction datasets [8, 9] improves their overall generalization, performance on specific tasks often remains suboptimal [7]. This raises a key challenge: *how can we efficiently select data subsets from general instruction datasets to maximize LLM performance on targeted tasks, given only a handful of examples* (§ 2.1) [10, 11].

---

\*Corresponding author

<sup>2</sup>Available on <https://github.com/OpenDFM/MoNA>

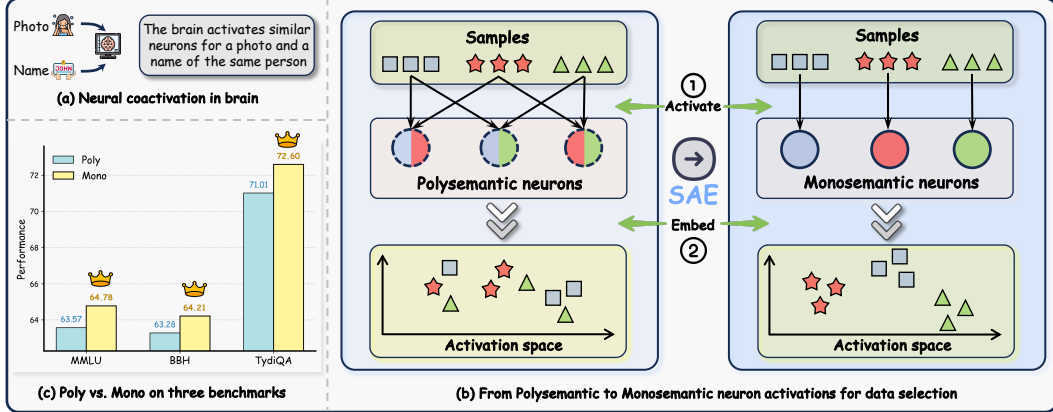


Figure 1: (a) An example of neural coactivation in brain [25] (b) Disentangling polysemantic activations into monosemantic representations via a sparse autoencoder (SAE) (c) Improved data selection using monosemantic activations (More details are in Appendix B.3)

To address this challenge, data selection methods can be roughly grouped into two main categories: *influence-based* methods and *distribution alignment* methods. Influence-based methods [11–13] select training data by estimating the influence of each candidate sample on the evaluation loss for target examples [14], and prioritize samples that are expected to most reduce this loss. However, these approaches can be *unstable* since evaluation loss can fail to reflect true model performance after instruction tuning, causing inconsistency in selection results [15–17]. As an alternative, distribution alignment methods [18–21] select data whose distribution is similar to that of the target task, in order to minimize distribution shift and improve generalization [10, 22–24]. These methods typically embed samples into a feature space, define a similarity metric within this space, and select data most similar to representative samples of the target task (§ 2.2).

While distribution alignment offers a principled framework for data selection, its effectiveness depends heavily on how samples are represented. Existing methods typically rely on *data-centric* embeddings, derived either from shallow textual features (e.g., BM25 [18] and DSIR [19]) or high-dimensional neural representations (e.g., BGE [20] and LLM2Vec [21]) extracted directly from the data. While effective in many settings, such representations may fail to capture the internal computational dynamics of how the model processes different data points, which is crucial for task-specific performance [26, 27]. To address this, we introduce a *model-centric* paradigm which explicitly captures these internal dynamics by *representing each data point through the neuronal activation pattern it triggers in a pretrained model*. This approach is inspired by neuroscience, where related concepts trigger coordinated neural responses [28] (Figure 1-a). Accordingly, we select those training samples whose activation signatures in the model are most similar to those elicited by target task exemplars.

Nonetheless, directly using raw neuron activations can yield suboptimal similarity estimates due to neuron polysemanticity, where a single neuron may respond to multiple, unrelated concepts<sup>3</sup> [26]. As a consequence, unrelated samples can appear spuriously similar due to shared activation of polysemantic neurons (left of Figure 1-b). To mitigate this issue, we employ sparse autoencoders (SAEs) [27, 29, 30] to disentangle polysemantic activations into sparse, monosemantic units, making similarity in activation space more aligned with semantic similarity (right of Figure 1-b, § 2.3). This improvement is illustrated in Figure 1-c, where data selection based on monosemantic activations yields superior performance compared to raw polysemantic activations. In addition, we introduce a dedicated similarity metric tailored to the sparse monosemantic activation space produced by SAEs, as detailed in § 2.4.

In summary, our contributions are: 1) We propose **MONA** (**Monosemantic Neuronal Activation-based Data Selection**), a novel method for task-specific data selection in instruction tuning. MONA represents each sample using sparse, monosemantic neuronal activations derived from sparse autoencoders,

<sup>3</sup>For example, one neuron in a large language model may be activated by both academic citations and HTTP request patterns.

enabling model-centric data selection. 2) We introduce a similarity metric tailored to this sparse monosemantic activation space, enabling more accurate identification of task-relevant examples. 3) Comprehensive experiments across multiple candidate instruction tuning datasets, evaluation tasks, models, and data selection ratios demonstrate that MONA consistently outperforms existing data selection approaches. We will release our code to facilitate further research in the community.

## 2 Methods

We begin this section by formalizing the task-specific data selection problem and its objective. Next, we outline the distribution alignment framework that underpins our approach. We then describe our proposed method (MONA) in detail, including the construction of monosemantic neuronal activation embeddings and the dedicated similarity metric designed for this embedding space.

### 2.1 Problem Formulation

Given a large-scale general instruction dataset  $\mathcal{D}^{\text{src}} = \{\mathbf{s}_i\}_{i=1}^N$  and a small set of representative examples from the target task  $\mathcal{D}^{\text{tgt}} = \{\mathbf{s}_j\}_{j=1}^M$ , where  $M \ll N$ , our goal is to select a subset  $\mathcal{D}^{\text{sel}} \subset \mathcal{D}^{\text{src}}$  such that fine-tuning a large language model (LLM) on  $\mathcal{D}^{\text{sel}}$  leads to the best overall performance on the target task, denoted as  $\mathcal{T}^{\text{tgt}}$ .

Formally, let  $\mathcal{M}_\theta$  denote an LLM with parameters  $\theta$ , and  $S(\mathcal{M}_\theta, \mathcal{T}^{\text{tgt}})$  denote its performance metric (e.g., accuracy, F1-score) on the target task. The data selection objective is formulated as:

$$\mathcal{D}^{\text{sel}} = \arg \max_{\mathcal{D} \subset \mathcal{D}^{\text{src}}, |\mathcal{D}|=k} S(\mathcal{M}_{\theta^*}, \mathcal{T}^{\text{tgt}}), \quad (1)$$

where  $|\mathcal{D}| = k$  is a budget constraint on the number of selected samples, and  $\theta^*$  is obtained by fine-tuning the model on  $\mathcal{D}$ :

$$\theta^* = \text{Optimize}(\mathcal{M}_\theta, \mathcal{D}). \quad (2)$$

However, directly optimizing Eq. (1) is computationally infeasible, as it involves enumerating all possible subsets and retraining the model for each. This motivates the need for efficient data selection criteria that can identify high-quality subsets with minimal supervision and computation.

### 2.2 Distribution Alignment Pipeline

A common approach to the data selection problem defined in § 2.1 is *distribution alignment*, which seeks to select a subset  $\mathcal{D}^{\text{sel}}$  whose distribution closely matches that of the target examples  $\mathcal{D}^{\text{tgt}}$ . The objective in Eq. (1) then becomes:

$$\mathcal{D}^{\text{sel}} = \arg \max_{\mathcal{D} \subset \mathcal{D}^{\text{src}}, |\mathcal{D}|=k} \text{Sim}(\mathcal{D}, \mathcal{D}^{\text{tgt}}), \quad (3)$$

where  $\text{Sim}(\cdot, \cdot)$  measures the distribution similarity between the selected and target sets. In practice, this similarity is operationalized by first computing, for each sample in  $\mathcal{D}$ , its similarity to the target set  $\mathcal{D}^{\text{tgt}}$  in the embedding space, and then aggregating these sample-level similarities. Based on this, the pipeline consists of three steps (see the left panel of Figure 2):

1. *Embedding*: Each sample  $\mathbf{s}_i \in \mathcal{D}^{\text{src}} \cup \mathcal{D}^{\text{tgt}}$  is projected into a  $d$ -dimensional feature space via an embedding function  $\Phi: \mathbf{s} \mapsto \mathbf{z} \in \mathbb{R}^d$ , where  $\mathbf{z}_i$  denotes the embedding of  $\mathbf{s}_i$ .<sup>4</sup>
2. *Similarity Metric Definition*: Explicitly define a similarity metric, denoted as  $\delta(\mathbf{s}_i, \mathcal{D}^{\text{tgt}})$ , to quantify the similarity between a source sample  $\mathbf{s}_i \in \mathcal{D}^{\text{src}}$  and the target set  $\mathcal{D}^{\text{tgt}}$ .
3. *Subset Selection*: The final subset  $\mathcal{D}^{\text{sel}}$  consists of the  $k$  samples from  $\mathcal{D}^{\text{src}}$  with the largest aggregate similarity to the target set:

$$\mathcal{D}^{\text{sel}} = \arg \max_{\mathcal{D} \subset \mathcal{D}^{\text{src}}, |\mathcal{D}|=k} \sum_{\mathbf{s}_i \in \mathcal{D}} \delta(\mathbf{s}_i, \mathcal{D}^{\text{tgt}}). \quad (4)$$

<sup>4</sup>In this paper,  $\mathbf{z}_i$  refers to the embedding of sample  $\mathbf{s}_i$ .

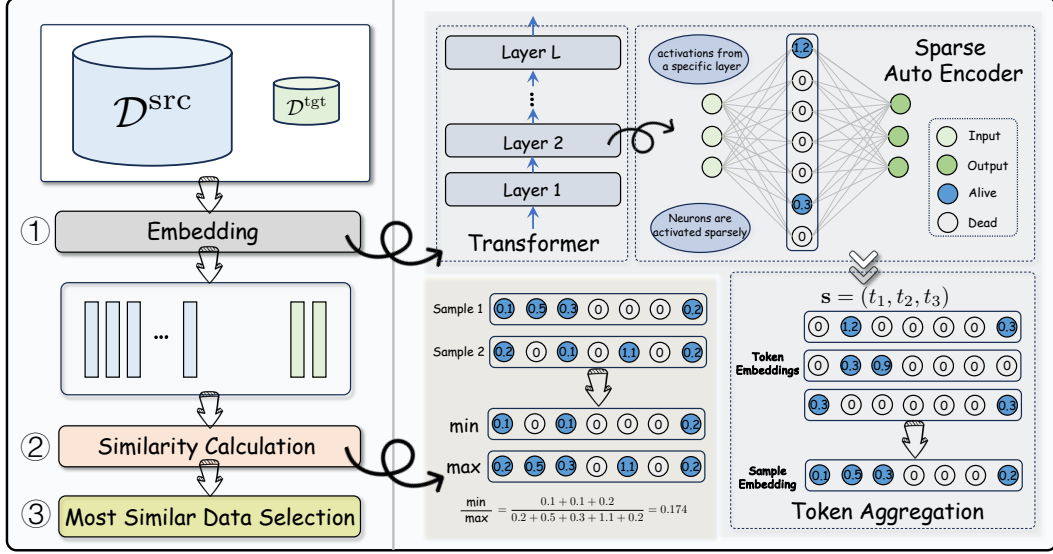


Figure 2: Workflow of MoNA. **Left:** Distribution alignment pipeline between the source dataset and the target task. **Right:** Computation of monosemantic neuronal activation embeddings and the proposed similarity metric. *Top:* Application of SAE; *Bottom right:* Aggregation of token embeddings into a sentence-level embedding; *Bottom left:* Calculation of similarity between two samples

### 2.3 Monosemantic Neuronal Activation Embedding

**Intuition** In neuroscience, related stimuli trigger coordinated neuron activations, reflecting semantic similarity [28]. Inspired by this, we hypothesize that neural network activations can similarly capture semantic relationships between samples. To this end, we represent each sample by its activation pattern from a *predefined set of neurons* (e.g., a specific layer)<sup>5</sup> in a neural network  $\mathcal{M}^{\text{ds}}$ :

$$\mathbf{z} = f_{\text{NAS}}(\mathcal{M}^{\text{ds}}, \mathbf{s}), \quad (5)$$

where  $f_{\text{NAS}}(\cdot, \cdot)$  extracts neuronal activations for input  $\mathbf{s}$  from  $\mathcal{M}^{\text{ds}}$ .

**Sparse Autoencoder-based Monosemantic Decomposition** Building on the intuition, we use the neuronal activation states from a single, predefined layer in the transformer [31] as the basis for the embedding.<sup>6</sup> However, even the activations from a single layer can exhibit polysemanticity, where a single neuron responds to multiple, often unrelated, concepts [26]. For example, one neuron in a large language model may be activated by both academic citations and HTTP request patterns, making such activations difficult to interpret and less effective for representing a specific semantic property [26]. This polysemanticity undermines the interpretability and reliability of feature representations based directly on raw activations.

To address this, we employ a sparse autoencoder (SAE) following [27, 29, 30] (see the top right part of Figure 2). The SAE transforms the original neuron activations into a higher-dimensional, sparse activation space, where each resulting neuron tends to respond to a distinct, monosemantic feature or concept, rather than simply producing a learned representation. Prior work [29] has demonstrated that such sparse activations exhibit improved interpretability and semantic purity, which is beneficial for our data selection framework.<sup>7</sup> Formally, given an input sequence  $\mathbf{s} = (t_1, t_2, \dots, t_n)$  of  $n$  tokens, let  $\mathbf{h}_k^\ell \in \mathbb{R}^d$  denote the output at layer  $\ell$  of model  $\mathcal{M}^{\text{ds}}$  for token  $t_k$  ( $1 \leq k \leq n$ ). The SAE computes a new sparse activation for each token as follows:

$$\mathbf{z}_k^\ell = \text{TopK}(\mathbf{W}_{\text{enc}}(\mathbf{h}_k^\ell - \mathbf{b}_{\text{pre}})), \quad (6)$$

<sup>5</sup>Using all neurons in the network would result in an extremely high-dimensional embedding, equal to the total number of model parameters, and bring significant computational overhead.

<sup>6</sup>We compare the effect of different layer choices in § 3.5.

<sup>7</sup>We provide further evidence for this through the visual validation shown in Figure 4.

where  $\mathbf{W}_{\text{enc}} \in \mathbb{R}^{d' \times d}$  ( $d' \gg d$ ) and  $\mathbf{b}_{\text{pre}} \in \mathbb{R}^d$  are trainable parameters. The operator  $\text{Top}K(\cdot)$  retains only the largest  $K$  values of the input vector and sets all remaining entries to zero.<sup>8</sup> This transformation produces a sparse and interpretable activation pattern for each token in the sequence, enabling more reliable and semantically meaningful representations for downstream data selection.

**Token Aggregation for Sample Embedding** After obtaining the sparse monosemantic activation  $\mathbf{z}_k^\ell$  for each token in the input, we aggregate these token-level vectors by averaging over all tokens to form a sample-level embedding (see the bottom right of Figure 2):

$$\mathbf{z} = \frac{1}{n} \sum_{k=1}^n \mathbf{z}_k^\ell, \quad (7)$$

where  $n$  is the number of tokens in the input  $\mathbf{s}$ . Averaging, rather than summing, is crucial for mitigating length bias: without normalization, the selection process systematically prefers samples that match the average length of samples in  $\mathcal{D}^{\text{tgt}}$ , rather than those with the highest semantic relevance. This bias often harms downstream performance, as demonstrated in Appendix D.5.

In summary,  $f_{\text{NAS}}(\cdot, \cdot)$  in Eq. (5) corresponds to the composition of the token-level sparse mapping in Eq. (6) and the aggregation operation in Eq. (7).

## 2.4 Similarity Metric for Monosemantic Neuronal Activation Embedding

This section describes how we define the similarity metric within the monosemantic neuronal activation embedding space for use in the distribution alignment pipeline. The procedure consists of two steps: (i) aggregating the embeddings of the target examples to form a task prototype, and (ii) computing the generalized Jaccard similarity [32] between each source sample and the task prototype.

**Task prototype representation** To improve efficiency, we aggregate the monosemantic neuronal activation embeddings of all target examples in  $\mathcal{D}^{\text{tgt}}$  into a single task prototype. This reduces the computational complexity from  $\mathcal{O}(|\mathcal{D}^{\text{tgt}}| \cdot |\mathcal{D}^{\text{src}}|)$  to  $\mathcal{O}(|\mathcal{D}^{\text{src}}|)$ . Formally, the task prototype is defined as:

$$\mathbf{z}^{\text{tgt}} = \frac{1}{|\mathcal{D}^{\text{tgt}}|} \sum_{\mathbf{s}_j \in \mathcal{D}^{\text{tgt}}} \mathbf{z}_j, \quad (8)$$

where  $\mathbf{z}_j$  is the embedding of the  $j$ -th target example.

**Generalized Jaccard Similarity** For high-dimensional, sparse feature representations such as our monosemantic neuronal activation embeddings, classic similarity metrics such as Euclidean or Cosine similarity can become unreliable due to the ‘‘curse of dimensionality’’ [33]. Our empirical results confirm that neither Euclidean nor Cosine similarity is suitable for this embedding space (see § 3.5 for details). As an alternative, we adopt the generalized Jaccard similarity (see the bottom left part of Figure 2). Mathematically, given a source sample  $\mathbf{s}_i$  with embedding  $\mathbf{z}_i$  and the task prototype  $\mathbf{z}^{\text{tgt}}$ , the generalized Jaccard similarity is defined as:

$$\delta(\mathbf{s}_i, \mathcal{D}^{\text{tgt}}) = \frac{\sum_k \min(\mathbf{z}_i[k], \mathbf{z}^{\text{tgt}}[k])}{\sum_k \max(\mathbf{z}_i[k], \mathbf{z}^{\text{tgt}}[k])}, \quad (9)$$

where  $\mathbf{z}_i[k]$  (or  $\mathbf{z}^{\text{tgt}}[k]$ ) denotes the  $k$ -th element of the corresponding embedding.

## 3 Experiments

In this section, we design experiments to systematically evaluate our method (MONA) for task-specific instruction tuning. We center our evaluation around the following key questions:

- **Effectiveness and Robustness (Q1):** Does MONA consistently select data that yields better downstream performance across (i) various source general instruction datasets and target evaluation tasks, (ii) different instruction-tuned LLMs, and (iii) a range of data selection ratios?

<sup>8</sup>We provide an ablation study of the effect of different  $K$  values in § 3.5.

Table 1: Performance of different models after instruction tuning with **5%** of the data selected from different datasets. **Best** results are in bold; second best are underlined.

Method	$\mathcal{D}^{\text{src}} = \text{OPENHERMES-2.5}$						$\mathcal{D}^{\text{src}} = \text{LESS}$			
	MMLU	GSM8K	BBH	MBPP	GPQA	Avg.	MMLU	BBH	TyDiQA	Avg.
<b>LLaMA3.1-8B</b>										
BASE	65.30	55.50	63.08	46.40	28.12	51.68	65.30	63.08	71.26	66.55
FULL	64.60	65.35	64.31	49.00	27.90	54.23	64.60	64.31	72.66	67.19
RANDOM	64.02	58.65	63.70	46.73	30.36	52.69	64.16	<b>64.29</b>	69.78	<u>66.08</u>
<i>Influence-based</i>										
MATES	64.11	54.28	65.38	47.60	28.12	51.90	63.62	63.68	67.74	65.01
LESS	64.34	66.87	63.00	47.80	<b>31.47</b>	54.70	62.51	62.11	70.68	65.10
<i>Distribution alignment</i>										
BM25	64.14	66.64	65.23	48.40	27.90	54.46	<u>64.41</u>	63.74	68.07	65.41
DSIR	63.95	<u>66.94</u>	64.29	<u>48.60</u>	29.91	<u>54.74</u>	64.25	63.19	65.61	64.35
DLRDS-BGE	<u>64.45</u>	64.82	64.20	<u>48.60</u>	<u>31.25</u>	54.66	64.06	61.82	70.30	65.39
DLRDS-LLaMA3-8B	64.31	64.75	63.97	<b>48.80</b>	29.46	54.26	62.11	61.54	<u>71.91</u>	65.19
LLM2Vec	64.29	63.53	<u>65.55</u>	48.40	30.13	54.38	62.06	62.03	68.11	64.07
MoNA (ours)	<b>64.49</b>	<b>67.93</b>	<b>66.44</b>	48.40	<b>31.47</b>	<b>55.75</b>	<b>64.78</b>	<u>64.21</u>	<b>72.60</b>	<b>67.20</b>
<b>OLMo-7B</b>										
BASE	28.42	7.35	29.96	21.40	26.56	22.74	28.42	29.96	31.67	30.02
FULL	45.05	31.96	33.13	26.40	26.56	32.62	39.31	28.86	33.43	33.87
RANDOM	<u>36.96</u>	<u>16.00</u>	<u>31.47</u>	<u>19.47</u>	<u>27.38</u>	26.26	<u>28.60</u>	<u>30.82</u>	<u>31.93</u>	30.45
<i>Influence-based</i>										
MATES	30.27	13.72	32.33	16.40	27.01	23.95	29.57	30.46	31.02	30.35
LESS	<b>46.15</b>	26.91	33.68	20.20	25.89	30.57	37.21	30.07	33.20	33.49
<i>Distribution alignment</i>										
BM25	42.34	31.08	<b>34.30</b>	<b>26.80</b>	25.45	<u>31.99</u>	35.74	28.95	<b>34.40</b>	33.03
DSIR	36.48	29.26	<u>34.08</u>	19.40	27.23	29.29	29.54	<b>32.87</b>	33.25	31.89
DLRDS-BGE	42.77	<u>32.30</u>	33.40	<b>26.80</b>	23.88	31.83	35.22	25.65	33.28	31.38
DLRDS-LLaMA3-8B	38.16	31.39	33.30	22.80	<b>30.13</b>	31.16	<b>40.64</b>	26.08	31.08	32.60
LLM2Vec	37.24	30.10	33.57	23.40	<u>28.35</u>	30.53	39.72	28.58	32.26	<u>33.52</u>
MoNA (ours)	<u>44.74</u>	<b>32.83</b>	33.51	<u>26.00</u>	25.00	<b>32.42</b>	<u>40.14</u>	30.19	<u>33.80</u>	<b>34.71</b>

- **Visualization and Interpretability (Q2):** Can the monosemantic activation embeddings make data selection decisions more transparent and interpretable? We illustrate this via visual analysis of activation patterns.
- **Key Factor Analysis (Q3):** How do crucial factors—such as layer selection, sparsity parameter  $K$ , and similarity metric—affect the behavior of MoNA?

### 3.1 Experimental Setup

**General Instruction Data and Evaluation Tasks** To comprehensively evaluate robustness and generalization on target tasks, we select training data from two large-scale, diverse instruction datasets: OPENHERMES-2.5 [9] (1M synthetic and curated instruction/chat samples) and LESS [11] (270K samples covering both classical sources such as FLAN V2 [8], CoT [34], and open-ended human-annotated datasets like DOLLY [35] and OPEN ASSISTANT 1 [36]). We evaluate performance on six target tasks: MMLU [37] (general knowledge), BBH [38] (complex reasoning), GSM8K [39] (math problems), MBPP [40] (programming), GPQA [41] (expert QA), and TyDiQA [42] (multilingual QA). Evaluations use lm-evaluation-harness [43] and vLLM [44] except for TyDiQA, which uses the LESS codebase [11]. More details are in Appendix B.1.

**Models and Training** We conduct instruction tuning on two widely used open-source language models: LLaMA3.1-8B [5] and OLMo-7B [45], with additional experiments on a larger model, LLaMA2-13B [46], to assess scalability. For data selection, we utilize open-source sparse autoencoder (SAE) models based on LLaMA3-8B<sup>9</sup>, with setting  $K = 192$  (§ 2.3). Neuron activations are extracted from the penultimate (second-to-last) layer of the model. Fine-tuning is performed with llama-factory [47], using a cosine scheduler (peak learning rate  $7\text{e-}6$ , warmup ratio 0.01), batch size 128, weight decay 0.1, and maximum sequence length 8192. All models are trained for two epochs. More details are shown in Appendix A.

<sup>9</sup><https://huggingface.co/EleutherAI/sae-llama-3-8b-32x>

Table 2: Performance of LLaMA2-13B after instruction tuning with **5%** of the data selected from OPENHERMES-2.5. **Best** results are in bold; second best are underlined.

Method	MMLU	GSM8K	BBH	MBPP	GPQA	Avg.
BASE	55.11	24.03	46.74	27.00	30.58	36.69
FULL	57.61	55.95	52.63	35.00	27.01	45.64
RANDOM	55.96	41.50	51.37	31.13	<b>28.64</b>	41.72
<i>Influence-based</i>						
MATES	55.68	37.07	51.17	31.20	26.34	40.29
LESS	<b>60.38</b>	48.75	50.42	25.80	<u>27.90</u>	42.65
<i>Distribution alignment</i>						
BM25	57.60	58.15	<b>52.65</b>	34.60	<u>27.90</u>	<u>46.18</u>
DSIR	55.83	53.53	52.02	31.60	27.23	44.04
DLRDS-BGE	56.65	56.63	52.34	<b>35.60</b>	27.68	45.78
DLRDS-LLaMA3-8B	<u>58.65</u>	52.31	<u>52.36</u>	<b>35.20</b>	26.56	45.02
LLM2Vec	57.02	<u>58.30</u>	51.27	34.80	<u>27.90</u>	45.86
MoNA (ours)	57.26	<b>60.27</b>	52.23	<b>35.60</b>	<u>27.90</u>	<b>46.65</b>

### 3.2 Baselines

To ensure fair and comprehensive comparison, we evaluate MONA against several representative baselines covering all major categories from § 1: (i) *Non-selection baselines*: BASE (base pretrained model) and FULL (instruction tuning on full instruction data); (ii) *Random selection* (RANDOM): uniformly samples data for fine-tuning (averaged over three seeds); (iii) *Influence-based selection*: MATES [13] (proxy model predicts loss reduction per sample) and LESS [11] (gradient-based Taylor approximation estimates influence); (iv) *Distribution alignment-based selection*: includes classical methods such as BM25 [18] (tf-idf) and DSIR [19] (n-gram), as well as deep embedding approaches—DLRDS-BGE [20] (BGE embeddings), DLRDS-LLaMA3-8B [5] (LLaMA3-8B embeddings), and LLM2Vec [21] (bidirectional text encoders adapted from decoder-only LLMs). All baselines use the same data selection ratio for fair comparison; additional details are provided in Appendix C.

### 3.3 Main Results on Target Tasks

To address Q1 (effectiveness and robustness), we select 5% of the general instruction data for each data selection method, fine-tune two backbone models, and evaluate on target tasks. We also assess scalability by repeating the experiments with a larger model. In addition, we investigate the impact of varying the data selection ratio, and further employ an LLM-based data analyst to conduct a model-agnostic evaluation of the selected data quality.

**Across Datasets and Target Tasks** As shown in Table 1, on the LLaMA3.1-8B model, MONA achieves either the best or second-best performance on nearly all tasks for both OPENHERMES-2.5 and LESS instruction datasets. For instance, on OPENHERMES-2.5, MONA achieves the highest scores on GSM8K, BBH, and GPQA, and obtains the best overall average—even surpassing full-data fine-tuning. Similar trends are observed on LESS. These results demonstrate that MONA not only selects more semantically relevant data than all baselines, but also maintains robust performance across a wide range of tasks and instruction data sources.

**Across Backbone Models** On OLMo-7B, MONA achieves the highest overall average performance among all methods for both OPENHERMES-2.5 and LESS instruction datasets (Table 1). Although task-level stability<sup>10</sup> decreases for all methods compared to LLaMA3.1-8B, MONA still achieves the highest proportion of top-two finishes—ranking in the top two on 3 out of 5 tasks for

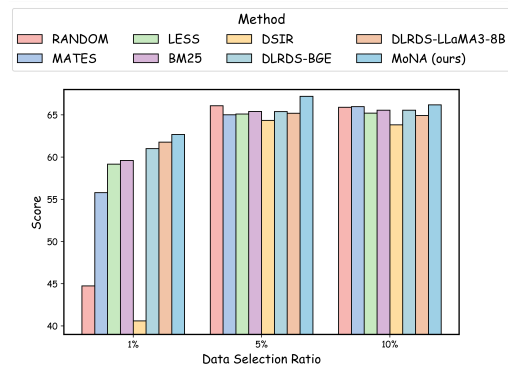


Figure 3: Performance of different data selection methods under varying selection ratios, evaluated on LESS with LLaMA3.1-8B.

<sup>10</sup>Here, task-level stability refers to the proportion of tasks where a method ranks among the top two. Lower stability means high performance is achieved on fewer tasks.



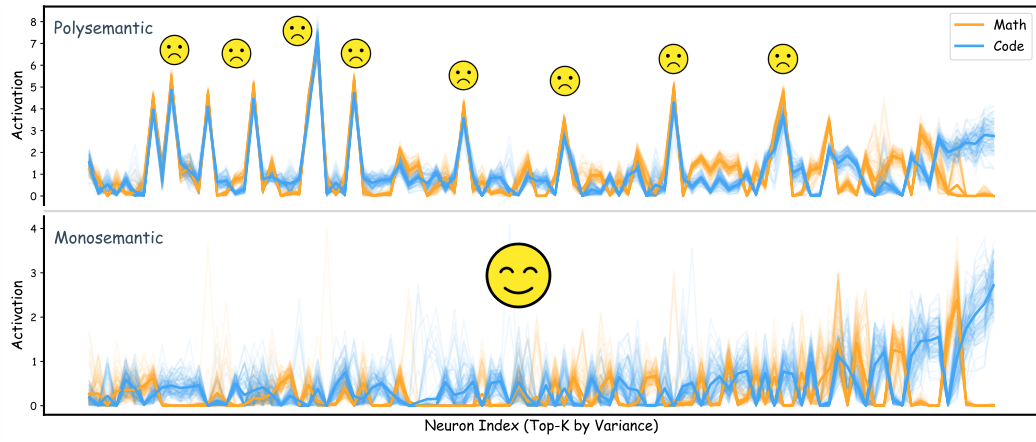


Figure 4: Neuron activation profiles for 100 Math and 100 Code samples on the top-100 most variant neurons. Faint lines show individual samples; bold lines show task means. In the polysemantic (top) plot, many neurons, especially those with high activation peaks (marked by weeping face), are simultaneously activated by both tasks, reflecting pronounced overlap and limited task specificity. In contrast, the monosemantic (bottom) plot reveals clear task-specific activation patterns.

OPENHERMES-2.5 and 2 out of 3 tasks for LESS, both of which are higher than any baseline. This indicates that, despite the absolute stability being affected by the backbone, MONA remains the most robust and semantically expressive data selection approach relative to competing methods.

On LLaMA2-13B, MONA exhibits a similar trend as observed on OLMo-7B (Table 2). Although absolute task-level stability decreases compared to LLaMA3.1-8B, MONA continues to show stronger overall performance and relatively higher stability than all baseline methods.

**Across Data Selection Ratios** Across all selection ratios (Figure 3), MONA consistently achieves the best performance, reaffirming both its robustness and semantic expressiveness. Interestingly, selecting 10% of the data results in lower performance than using 5%. We speculate that increasing the ratio may introduce less relevant or lower-quality samples, thereby diluting the benefits of high-quality, semantically aligned data. This observation indicates that the choice of selection ratio is a critical factor in data selection for instruction tuning and deserves further exploration.

**LLM as a Data Analyst** Beyond evaluating instruction-tuned model performance, we employ an LLM-based data analyst to assess the quality of selected training data in a model-agnostic way. For each method, we randomly sample 100 training instances and prompt GPT-4o-mini [48] to compare them with representative target samples, considering semantic similarity, instruction format, and task relevance. Each comparison is scored from 1 to 10, and final scores are averaged. More details are given in Appendix B.2. As shown in Figure 5, the LLM consistently assigns higher scores to data selected by MONA versus two strong baselines, confirming both the semantic expressiveness and stability of our approach. Additional case studies are provided in Appendix D.1.

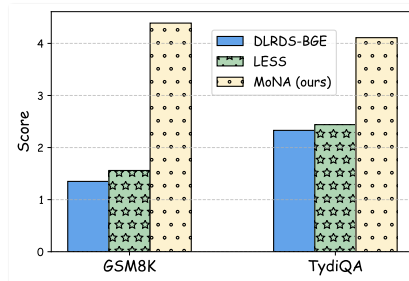


Figure 5: LLM as a Data Analyst: scores for data selected by different methods. Higher scores indicate better performance

### 3.4 Neuron Activation Visualization

In addition to validating improved downstream performance with neuron activation-based data selection, we further analyze and visualize the neuron activation patterns (Q2) for different tasks (Figure 4). While polysemantic neurons produce substantial activation overlap across tasks, monosemantic representations obtained via the sparse autoencoder yield well-separated task-specific activation patterns. This underscores the importance of disentangling polysemantic activations.



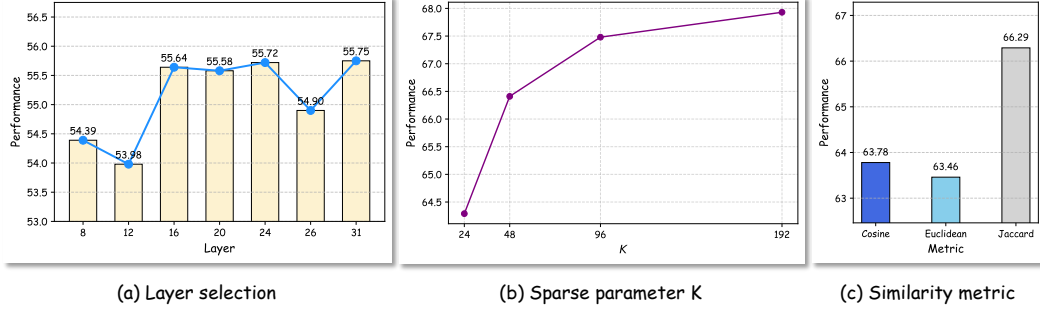


Figure 6: Ablation studies for key design choices in MoNA

### 3.5 Ablation Studies

To better understand the contribution of each component in MoNA (Q3), we conduct ablation studies on key design choices, including which layer to extract neuron activations from, the sparsity parameter  $K$ , and the similarity metric. More experimental details (e.g., evaluation benchmarks and instruction tuning datasets) are in Appendix D.8.

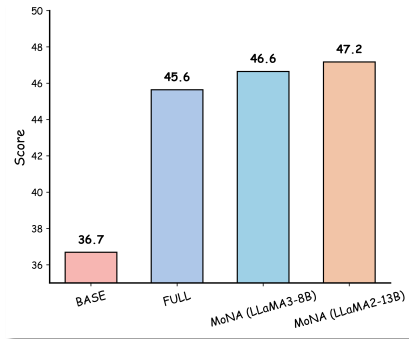
**Effect of Layer Selection** We first examine how the choice of layer for extracting neuron activations affects the performance of MoNA. Since prior work [30] shows that SAEs trained on shallower layers tend to specialize in next-token prediction and provide less transferable features, we focus our analysis on deeper layers. Specifically, we select seven layers evenly spaced from layer 8 to the penultimate layer (layer 31) of LLaMA3-8B. As shown in Figure 6-a, embeddings from shallower layers can result in unstable or suboptimal performance, while embeddings from deeper layers—especially the penultimate layer—deliver strong results. Based on these observations, we extract neuron activations from the penultimate layer in all other experiments.

**Effect of Sparsity Parameter  $K$**  As shown in Figure 6-b, model performance generally improves as the sparsity parameter  $K$  increases, indicating that retaining more active neurons leads to higher-quality data selection. The improvement becomes less pronounced as  $K$  grows larger. In contrast, when  $K$  is very small, performance drops sharply, suggesting that insufficient neuron information hampers effective selection. Based on these results, we adopt  $K = 192$  in all main experiments.<sup>11</sup>

**Effect of Similarity Metric** We compare the impact of different similarity metrics, including Jaccard, Cosine, and Euclidean. As shown in Figure 6-c, Jaccard similarity consistently yields better results than the other metrics across benchmarks, highlighting its suitability for MoNA.

### 3.6 Additional Analysis

**Effect of SAE** To examine the effect of the SAE model in greater depth, we additionally train an SAE model built upon LLaMA2-13B. The model is trained using the RedPajama-Data-1T-Sample dataset<sup>12</sup>, with training conducted under the sparsify framework<sup>13</sup>. As is shown in Figure 7, a larger SAE backbone can further enhance the effectiveness of data selection and downstream performance. These findings suggest that the quality and scale of the SAE model have a positive impact on the overall results.



**Comparison with MoE-based Monosemantic Embeddings** To further explore the concept of monosemantic embeddings, we investigate an MoE-based monosemantic embedding approach [49]. Specifically, we replace our SAE with MoE-based

Figure 7: Results of Using Different SAE Models on LLaMA2-13B

<sup>11</sup>We did not explore values of  $K$  greater than 192 in this work.

<sup>12</sup><https://huggingface.co/datasets/togethercomputer/RedPajama-Data-1T-Sample>

<sup>13</sup><https://github.com/EleutherAI/sparsify/tree/main>

embeddings: for each token, we use the concatenation of expert routing scores from all layers of an MoE model (collected during a forward pass) as the token’s embedding. For this experiment, we employ the allenai/OLMoE-1B-7B-0924 [50] model. Except for substituting the SAE-based embeddings with MoE-based embeddings, all other components and settings of our workflow remain unchanged to ensure a fair comparison.

Figure 8 demonstrates that using MoE-based embeddings yields a modest improvement over the FULL baseline under our current experimental setup, indicating that monosemantic representations from MoE models can provide a slight benefit for task-specific data selection. However, MoE-based embeddings still lag behind SAE-based embeddings by a noticeable margin. This observation suggests that the MoE-based embedding still has room for improvement in capturing monosemantic representations.

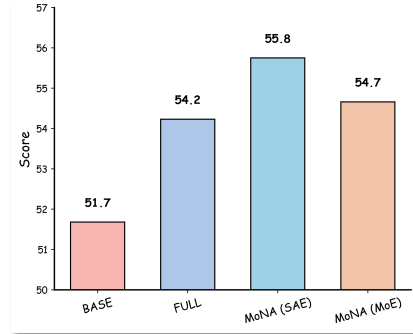


Figure 8: Evaluation of Monosemantic Embeddings over LLaMA3.1-8B: SAE-based vs. MoE-based

## 4 Related Work

### Data Selection for Task-Specific Instruction Tuning

Most data selection methods for task-specific instruction tuning roughly fall into two main categories: influence-based and distribution alignment approaches. Influence-based methods select training data by estimating the influence of each candidate sample on the evaluation loss for target examples [14], and prioritize samples expected to most reduce this loss. For example, DsDm [12] and MATES [13] use proxy models, while LESS [11] relies on gradients. However, these methods can be unstable because evaluation loss may not reliably indicate model quality after instruction tuning [15–17]. Distribution alignment methods [18–21] select samples by embedding them into a feature space and aligning the source and target distributions using a similarity metric. Prior work is largely data-centric, representing samples with surface-level features such as n-grams [19], tf-idf [18], or neural embeddings [20], which may fail to reflect how the model internally processes information [26, 27]. In contrast, our approach is model-centric: we represent each sample by the neuronal activation pattern it triggers inside the model, thereby capturing the internal computational dynamics that are critical for task-specific performance.

**Sparse Autoencoders in Feature Learning** Sparse coding was first introduced for over-complete dictionaries [51], and unsupervised dictionary learning was pioneered by [52]. These ideas led to sparse autoencoders, which have become important tools for learning structured features in vision and language [53, 54]. Recent work has also investigated sparse autoencoders in analyzing representations of large language models [26, 55]. Almost concurrently, [56] used sparse autoencoders for data selection, with a focus on diversity. In contrast, our approach is distinct in that we use sparse neuron activations to capture semantic relatedness between samples and to overcome neuron polysemanticity, enabling more interpretable and semantically aligned task-specific data selection.

## 5 Conclusion

We propose a model-centric approach for task-specific data selection in instruction tuning, using monosemantic neuronal activations from sparse autoencoders. This representation captures internal model computation and enables more semantically aligned and interpretable data selection. Experiments across various models and tasks demonstrate consistent gains over previous baselines.

## Acknowledgments

This work was supported by the China NSFC Projects (92370206, U23B2057, 62120106006), and the Shanghai Municipal Science and Technology Projects (2021SHZDZX0102 and 25X010202846).

## References

- [1] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray *et al.*, “Training language models to follow instructions with human feedback,” *Advances in neural information processing systems*, vol. 35, pp. 27 730–27 744, 2022.
- [2] S. Zhang, L. Dong, X. Li, S. Zhang, X. Sun, S. Wang, J. Li, R. Hu, T. Zhang, F. Wu *et al.*, “Instruction tuning for large language models: A survey,” *arXiv preprint arXiv:2308.10792*, 2023.
- [3] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, “Language models are few-shot learners,” *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.
- [4] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat *et al.*, “Gpt-4 technical report,” *arXiv preprint arXiv:2303.08774*, 2023.
- [5] A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan *et al.*, “The llama 3 herd of models,” *arXiv preprint arXiv:2407.21783*, 2024.
- [6] Anthropic, “Introducing the next generation of claude,” 2024. [Online]. Available: <https://www.anthropic.com/news/claude-3-family>
- [7] Y. Wang, H. Ivison, P. Dasigi, J. Hessel, T. Khot, K. Chandu, D. Wadden, K. MacMillan, N. A. Smith, I. Beltagy *et al.*, “How far can camels go? exploring the state of instruction tuning on open resources,” *Advances in Neural Information Processing Systems*, vol. 36, pp. 74 764–74 786, 2023.
- [8] S. Longpre, L. Hou, T. Vu, A. Webson, H. W. Chung, Y. Tay, D. Zhou, Q. V. Le, B. Zoph, J. Wei *et al.*, “The flan collection: Designing data and methods for effective instruction tuning,” in *International Conference on Machine Learning*. PMLR, 2023, pp. 22 631–22 648.
- [9] Teknium, “Openhermes 2.5: An open dataset of synthetic data for generalist llm assistants,” 2023. [Online]. Available: <https://huggingface.co/datasets/teknium/OpenHermes-2.5>
- [10] Z. Liu, A. Karbasi, and T. Rekatsinas, “TSDS: Data selection for task-specific model finetuning,” in *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. [Online]. Available: <https://openreview.net/forum?id=wjbTHLUSzU>
- [11] M. Xia, S. Malladi, S. Gururangan, S. Arora, and D. Chen, “Less: selecting influential data for targeted instruction tuning,” in *Proceedings of the 41st International Conference on Machine Learning*, ser. ICML’24. JMLR.org, 2025.
- [12] L. Engstrom, A. Feldmann, and A. Mkadry, “Dsdm: model-aware dataset selection with datamodels,” in *Proceedings of the 41st International Conference on Machine Learning*, ser. ICML’24. JMLR.org, 2025.
- [13] Z. Yu, S. Das, and C. Xiong, “MATES: Model-aware data selection for efficient pretraining with data influence models,” in *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. [Online]. Available: <https://openreview.net/forum?id=6gzPSMUz2>
- [14] T. Thrush, C. Potts, and T. Hashimoto, “Improving pretraining data using perplexity correlations,” *arXiv preprint arXiv:2409.05816*, 2024.
- [15] Y. Tay, M. Dehghani, J. Rao, W. Fedus, S. Abnar, H. W. Chung, S. Narang, D. Yogatama, A. Vaswani, and D. Metzler, “Scale efficiently: Insights from pretraining and finetuning transformers,” in *International Conference on Learning Representations*, 2022. [Online]. Available: <https://openreview.net/forum?id=f2OYVDyfiB>
- [16] C. Zhou, P. Liu, P. Xu, S. Iyer, J. Sun, Y. Mao, X. Ma, A. Efrat, P. Yu, L. Yu, S. Zhang, G. Ghosh, M. Lewis, L. Zettlemoyer, and O. Levy, “Lima: less is more for alignment,” in *Proceedings of the 37th International Conference on Neural Information Processing Systems*, ser. NIPS ’23. Red Hook, NY, USA: Curran Associates Inc., 2024.
- [17] T. Xia, B. Yu, K. Dang, A. Yang, Y. Wu, Y. Tian, Y. Chang, and J. Lin, “Rethinking data selection at scale: Random selection is almost all you need,” *arXiv preprint arXiv:2410.09335*, 2024.
- [18] S. Robertson, H. Zaragoza *et al.*, “The probabilistic relevance framework: Bm25 and beyond,” *Foundations and Trends® in Information Retrieval*, vol. 3, no. 4, pp. 333–389, 2009.
- [19] S. M. Xie, S. Santurkar, T. Ma, and P. S. Liang, “Data selection for language models via importance resampling,” *Advances in Neural Information Processing Systems*, vol. 36, pp. 34 201–34 227, 2023.

- [20] S. Xiao, Z. Liu, P. Zhang, N. Muennighoff, D. Lian, and J.-Y. Nie, “C-pack: Packed resources for general chinese embeddings,” in *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2024, pp. 641–649.
- [21] P. BehnamGhader, V. Adlakha, M. Mosbach, D. Bahdanau, N. Chapados, and S. Reddy, “LLM2vec: Large language models are secretly powerful text encoders,” in *First Conference on Language Modeling*, 2024. [Online]. Available: <https://openreview.net/forum?id=IW1PR7vEBf>
- [22] C. Jia and Y. Zhang, “Prompt-based distribution alignment for domain generalization in text classification,” in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Y. Goldberg, Z. Kozareva, and Y. Zhang, Eds. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 10 147–10 157. [Online]. Available: <https://aclanthology.org/2022.emnlp-main.690/>
- [23] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, “A theory of learning from different domains,” *Machine learning*, vol. 79, pp. 151–175, 2010.
- [24] A. Torralba and A. A. Efros, “Unbiased look at dataset bias,” in *CVPR 2011*. IEEE, 2011, pp. 1521–1528.
- [25] R. Q. Quiroga, L. Reddy, G. Kreiman, C. Koch, and I. Fried, “Invariant visual representation by single neurons in the human brain,” *Nature*, vol. 435, no. 7045, pp. 1102–1107, 2005.
- [26] T. Bricken, A. Templeton, J. Batson, B. Chen, A. Jermyn, T. Conerly, N. Turner, C. Anil, C. Denison, A. Askell, R. Lasenby, Y. Wu, S. Kravec, N. Schiefer, T. Maxwell, N. Joseph, Z. Hatfield-Dodds, A. Tamkin, K. Nguyen, B. McLean, J. E. Burke, T. Hume, S. Carter, T. Henighan, and C. Olah, “Towards monosemanticity: Decomposing language models with dictionary learning,” *Transformer Circuits Thread*, 2023, <https://transformer-circuits.pub/2023/monosemantic-features/index.html>.
- [27] R. Huben, H. Cunningham, L. R. Smith, A. Ewart, and L. Sharkey, “Sparse autoencoders find highly interpretable features in language models,” in *The Twelfth International Conference on Learning Representations*, 2024. [Online]. Available: <https://openreview.net/forum?id=F76bwRSLek>
- [28] J. Taubert, V. Goffaux, G. Van Belle, W. Vanduffel, and R. Vogels, “The impact of orientation filtering on face-selective neurons in monkey inferior temporal cortex,” *Scientific reports*, vol. 6, no. 1, p. 21189, 2016.
- [29] A. Templeton, T. Conerly, J. Marcus, J. Lindsey, T. Bricken, B. Chen, A. Pearce, C. Citro, E. Ameisen, A. Jones, H. Cunningham, N. L. Turner, C. McDougall, M. MacDiarmid, C. D. Freeman, T. R. Sumers, E. Rees, J. Batson, A. Jermyn, S. Carter, C. Olah, and T. Henighan, “Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet,” *Transformer Circuits Thread*, 2024. [Online]. Available: <https://transformer-circuits.pub/2024/scaling-monosemanticity/index.html>
- [30] L. Gao, T. D. la Tour, H. Tillman, G. Goh, R. Troll, A. Radford, I. Sutskever, J. Leike, and J. Wu, “Scaling and evaluating sparse autoencoders,” in *The Thirteenth International Conference on Learning Representations*, 2025. [Online]. Available: <https://openreview.net/forum?id=tcsZt9ZNKD>
- [31] A. Vaswani, “Attention is all you need,” *Advances in Neural Information Processing Systems*, 2017.
- [32] A. Huang *et al.*, “Similarity measures for text document clustering,” in *Proceedings of the sixth new zealand computer science research student conference (NZCSRSC2008)*, Christchurch, New Zealand, vol. 4, 2008, pp. 9–56.
- [33] C. Aggarwal, “Data mining the text book,” 2015.
- [34] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. H. Chi, Q. V. Le, and D. Zhou, “Chain-of-thought prompting elicits reasoning in large language models,” in *Proceedings of the 36th International Conference on Neural Information Processing Systems*, ser. NIPS ’22. Red Hook, NY, USA: Curran Associates Inc., 2022.
- [35] M. Conover, M. Hayes, A. Mathur, J. Xie, J. Wan, S. Shah, A. Ghodsi, P. Wendell, M. Zaharia, and R. Xin, “Free dolly: Introducing the world’s first truly open instruction-tuned llm,” 2023. [Online]. Available: <https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm>
- [36] A. Köpf, Y. Kilcher, D. von Rütte, S. Anagnostidis, Z.-R. Tam, K. Stevens, A. Barhoum, N. M. Duc, O. Stanley, R. Nagyfi, S. ES, S. Suri, D. Glushkov, A. Dantuluri, A. Maguire, C. Schuhmann, H. Nguyen, and A. Mattick, “Openassistant conversations - democratizing large language model alignment,” in *Proceedings of the 37th International Conference on Neural Information Processing Systems*, ser. NIPS ’23. Red Hook, NY, USA: Curran Associates Inc., 2023.

- [37] D. Hendrycks, C. Burns, S. Basart, A. Zou, M. Mazeika, D. Song, and J. Steinhardt, “Measuring massive multitask language understanding,” in *International Conference on Learning Representations*, 2021. [Online]. Available: <https://openreview.net/forum?id=d7KBjmI3GmQ>
- [38] A. Srivastava, A. Rastogi, A. Rao, A. A. M. Shueb, A. Abid, A. Fisch *et al.*, “Beyond the imitation game: Quantifying and extrapolating the capabilities of language models,” *Transactions on Machine Learning Research*, 2023, featured Certification. [Online]. Available: <https://openreview.net/forum?id=uyTL5Bvosj>
- [39] K. Cobbe, V. Kosaraju, M. Bavarian, M. Chen, H. Jun, L. Kaiser, M. Plappert, J. Tworek, J. Hilton, R. Nakano, C. Hesse, and J. Schulman, “Training verifiers to solve math word problems,” *arXiv preprint arXiv:2110.14168*, 2021.
- [40] J. Austin, A. Odena, M. Nye, M. Bosma, H. Michalewski, D. Dohan, E. Jiang, C. Cai, M. Terry, Q. Le *et al.*, “Program synthesis with large language models,” *arXiv preprint arXiv:2108.07732*, 2021.
- [41] D. Rein, B. L. Hou, A. C. Stickland, J. Petty, R. Y. Pang, J. Dirani, J. Michael, and S. R. Bowman, “GPQA: A graduate-level google-proof q&a benchmark,” in *First Conference on Language Modeling*, 2024. [Online]. Available: <https://openreview.net/forum?id=Ti67584b98>
- [42] J. H. Clark, E. Choi, M. Collins, D. Garrette, T. Kwiatkowski, V. Nikolaev, and J. Palomaki, “TyDi QA: A benchmark for information-seeking question answering in typologically diverse languages,” *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 454–470, 2020. [Online]. Available: <https://aclanthology.org/2020.tacl-1.30/>
- [43] L. Gao, J. Tow, B. Abbasi, S. Biderman, S. Black, A. DiPofi, C. Foster, L. Golding, J. Hsu, A. Le Noac’h, H. Li, K. McDonell, N. Muennighoff, C. Ociepa, J. Phang, L. Reynolds, H. Schoelkopf, A. Skowron, L. Sutawika, E. Tang, A. Thite, B. Wang, K. Wang, and A. Zou, “A framework for few-shot language model evaluation,” 07 2024. [Online]. Available: <https://zenodo.org/records/12608602>
- [44] W. Kwon, Z. Li, S. Zhuang, Y. Sheng, L. Zheng, C. H. Yu, J. E. Gonzalez, H. Zhang, and I. Stoica, “Efficient memory management for large language model serving with pagedattention,” in *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.
- [45] D. Groeneveld, I. Beltagy, E. Walsh, A. Bhagia, R. Kinney, O. Tafjord, A. Jha *et al.*, “OLMo: Accelerating the science of language models,” in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, L.-W. Ku, A. Martins, and V. Srikumar, Eds. Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 15 789–15 809. [Online]. Available: <https://aclanthology.org/2024.acl-long.841/>
- [46] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale *et al.*, “Llama 2: Open foundation and fine-tuned chat models,” *arXiv preprint arXiv:2307.09288*, 2023.
- [47] Y. Zheng, R. Zhang, J. Zhang, Y. Ye, Z. Luo, Z. Feng, and Y. Ma, “Llamafactory: Unified efficient fine-tuning of 100+ language models,” in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*. Bangkok, Thailand: Association for Computational Linguistics, 2024. [Online]. Available: <http://arxiv.org/abs/2403.13372>
- [48] OpenAI, “Gpt-4 technical report,” *arXiv preprint arXiv:2303.08774*, 2023.
- [49] Z. Li and T. Zhou, “Your mixture-of-experts LLM is secretly an embedding model for free,” in *The Thirteenth International Conference on Learning Representations*, 2025. [Online]. Available: <https://openreview.net/forum?id=eFGQ97z5Cd>
- [50] N. Muennighoff, L. Soldaini, D. Groeneveld, K. Lo, J. Morrison, S. Min, W. Shi, E. P. Walsh, O. Tafjord, N. Lambert, Y. Gu, S. Arora, A. Bhagia, D. Schwenk, D. Wadden, A. Wettig, B. Hui, T. Dettmers, D. Kiela, A. Farhadi, N. A. Smith, P. W. Koh, A. Singh, and H. Hajishirzi, “OLMoe: Open mixture-of-experts language models,” in *The Thirteenth International Conference on Learning Representations*, 2025. [Online]. Available: <https://openreview.net/forum?id=xXTkbTBmq>
- [51] S. Mallat and Z. Zhang, “Matching pursuits with time-frequency dictionaries,” *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3397–3415, 1993.
- [52] B. A. Olshausen and D. J. Field, “Emergence of simple-cell receptive field properties by learning a sparse code for natural images,” *Nature*, vol. 381, no. 6583, pp. 607–609, 1996.
- [53] H. Lee, C. Ekanadham, and A. Ng, “Sparse deep belief net model for visual area v2,” *Advances in neural information processing systems*, vol. 20, 2007.

- [54] A. Makhzani and B. Frey, “K-sparse autoencoders,” *arXiv preprint arXiv:1312.5663*, 2013.
- [55] D. Mossing, S. Bills, H. Tillman, T. Dupré la Tour, N. Cammarata, L. Gao, J. Achiam, C. Yeh, J. Leike, J. Wu, and W. Saunders, “Transformer debugger,” <https://github.com/openai/transformer-debugger>, 2024.
- [56] X. Yang, S. Nie, L. Liu, S. Gururangan, U. Karn, R. Hou, M. Khabsa, and Y. Mao, “Diversity-driven data selection for language model tuning through sparse autoencoder,” *arXiv preprint arXiv:2502.14050*, 2025.
- [57] D. Lee, J. Lee, G. Zhang, M. Tiwari, and A. Mirhoseini, “CATS: Context-aware thresholding for sparsity in large language models,” in *First Conference on Language Modeling*, 2024. [Online]. Available: <https://openreview.net/forum?id=v3w2a7EInO>
- [58] S. Elfving, E. Uchibe, and K. Doya, “Sigmoid-weighted linear units for neural network function approximation in reinforcement learning,” *Neural Networks*, vol. 107, pp. 3–11, 2018, special issue on deep reinforcement learning. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0893608017302976>
- [59] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, J. Burstein, C. Doran, and T. Solorio, Eds. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. [Online]. Available: <https://aclanthology.org/N19-1423/>
- [60] X. H. Lù, “Bm25s: Orders of magnitude faster lexical search via eager sparse scoring,” 2024. [Online]. Available: <https://arxiv.org/abs/2407.03618>
- [61] M. Artetxe and H. Schwenk, “Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond,” *Transactions of the Association for Computational Linguistics*, vol. 7, pp. 597–610, 2019. [Online]. Available: <https://aclanthology.org/Q19-1038/>
- [62] M. Artetxe, S. Ruder, and D. Yogatama, “On the cross-lingual transferability of monolingual representations,” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, D. Jurafsky, J. Chai, N. Schlueter, and J. Tetreault, Eds. Online: Association for Computational Linguistics, Jul. 2020, pp. 4623–4637. [Online]. Available: <https://aclanthology.org/2020.acl-main.421/>
- [63] Z. Zhu, Y. Xu, L. Chen, J. Yang, Y. Ma, Y. Sun, H. Wen, J. Liu, J. Cai, Y. Ma *et al.*, “Multi: Multimodal understanding leaderboard with text and images,” *arXiv preprint arXiv:2402.03173*, 2024.
- [64] Y. Wang, Y. Kordi, S. Mishra, A. Liu, N. A. Smith, D. Khashabi, and H. Hajishirzi, “Self-instruct: Aligning language models with self-generated instructions,” in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, A. Rogers, J. Boyd-Graber, and N. Okazaki, Eds. Toronto, Canada: Association for Computational Linguistics, Jul. 2023, pp. 13 484–13 508. [Online]. Available: <https://aclanthology.org/2023.acl-long.754/>
- [65] P. M. McCarthy and S. Jarvis, “Mtld, vocd-d, and hd-d: A validation study of sophisticated approaches to lexical diversity assessment,” *Behavior Research Methods*, vol. 42, no. 2, pp. 381–392, 2010. [Online]. Available: <https://pubmed.ncbi.nlm.nih.gov/20479170>

## NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [NA].
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

**The checklist answers are an integral part of your paper submission.** They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes]" is generally preferable to "[No]", it is perfectly acceptable to answer "[No]" provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No]" or "[NA]" is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading “NeurIPS Paper Checklist”,**
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers.**

### 1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper’s contributions and scope?

Answer: [Yes]

Justification: The abstract and introduction accurately reflect the main contributions and scope of the paper. All claims—regarding the proposed model-centric, monosemantic activation-based data selection method, the new similarity metric, and the comprehensive evaluation—are directly addressed and validated in the subsequent sections.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

### 2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?



Answer: [\[Yes\]](#)

Justification: The limitations of this work are discussed in § 5.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

### 3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [\[Yes\]](#)

Justification: Formulas in the paper are numbered and referenced properly. Necessary brief theoretical justifications are in Appendix D.5.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

### 4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [\[Yes\]](#)

Justification: The paper provides all necessary details for reproducing the main experiments, including descriptions of datasets, model architectures, training procedures, hyperparameters, evaluation metrics, and data selection settings. All details are in § 3, Appendix A, Appendix B, Appendix C, and Appendix D.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

## 5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: Core components of our code are provided in the supplemental material. The full code with instructions will be released upon final organization.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.

- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

#### 6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [\[Yes\]](#)

Justification: All details can be found in § 3, Appendix A, Appendix B, Appendix C, and Appendix D.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

#### 7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [\[No\]](#)

Justification: Error bars are not reported because it would be too computationally expensive.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

#### 8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We provide the computer resources in Appendix A.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

## 9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines?>

Answer: [Yes]

Justification: The research conforms the NeurIPS Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

## 10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We discuss the broader impacts in § 5.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

## 11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The paper poses no such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

## 12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: This work is built upon open-source assets and we cited them properly.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, [paperswithcode.com/datasets](https://paperswithcode.com/datasets) has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

## 13. New assets

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: The paper does not release new assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

#### 14. Crowdsourcing and research with human subjects

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

#### 15. Institutional review board (IRB) approvals or equivalent for research with human subjects

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

#### 16. Declaration of LLM usage

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: Details are provided in § 3.3 and Appendix B.2.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.

## Impacts and Limitations

Our proposed data selection framework MONA offers a neuroscience-inspired perspective for improving task-specific instruction tuning. By enhancing the stability, semantic expressiveness, and interpretability of data selection, MONA has the potential to benefit a broad range of downstream language model applications and may inspire new research directions in data curation and model analysis, both in academia and industry.

While MONA demonstrates strong effectiveness for task-specific instruction tuning, its extension to other stages—such as pre-training data selection—as not been explored in this work. In addition, our current study focuses solely on the text modality. Applying MONA to multimodal data selection scenarios, for example image-text tasks, remains an open and promising direction for future research.

## A Training Details

We provide details on the input data formatting for instruction tuning across the three backbone models evaluated in our work: LLaMA3.1-8B, OLMo-7B, and LLaMA2-13B. For each model, we adopted the format recommended in its official documentation or open-source release. Below, we present a concrete data example for each model. Additionally, all experiments are conducted on NVIDIA A100, A800, and H800 GPUs.

### LLaMA3.1-8B

<|begin\_of\_text|><|start\_header\_id|>user<|end\_header\_id|>

A doctor gives you three pills. She tells you to take one every half hour. How long will the pills last?<|eot\_id|><|start\_header\_id|>assistant<|end\_header\_id|>

One hour. You take the first pill immediately, then the other two at half-hour intervals.<|eot\_id|>

### OLMo-7B

A doctor gives you three pills. She tells you to take one every half hour. How long will the pills last? One hour. You take the first pill immediately, then the other two at half-hour intervals.<|endoftext|>

### LLaMA2-13B

<s>[INST] A doctor gives you three pills. She tells you to take one every half hour. How long will the pills last? [/INST] One hour. You take the first pill immediately, then the other two at half-hour intervals. </s>

## B Evaluation Details

### B.1 Evaluation Tasks Details

For MMLU and MBPP, we directly use their respective validation sets as representative examples. For the remaining datasets without validation sets, we follow the strategies outlined below:

- GSM8K: We randomly select 100 samples from the training set to serve as representative examples.
- BBH: We extract representative examples by selecting the provided few-shot samples in the task setup.
- GPQA: We use the extended 98 data points, which are the "extended split" minus the "main split."
- TydiQA: Following [11], we select one sample per language as the representative example.



Table 3: Details of all evaluation tasks

Task	$ \mathcal{D}^{\text{tgt}} $	# Test Samples	Shot	Metric	Harness Task Name
MMLU	285	18,721	5	Accuracy	mmlu
GSM8K	100	2,638	8	Exact Match	gsm8k_cot
BBH	81	920	3	Exact Match	bbh
MBPP	90	500	3	pass@1	mbpp
GPQA	98	448	0	Accuracy	gpqa_main_zeroshot
TydiQA	9	5,077	1	F1	-

For more statistical information, evaluation metrics, and details regarding the LM-evaluation-harness setup, please refer to Table 3.

#### Prompt of LLM data analyst

You are an expert in evaluating task-specific data selection strategies for instruction tuning of large language models (LLMs). Your task is to assess how effectively the selected training data improve the performance of LLMs on a specific target task.

### Context:

1. You are provided with a **\*\*representative example\*\*** of the target task.
2. Only a small sample of the selected data is provided as a reference due to space limitations.
3. Your goal is to evaluate how well the selected data would help fine-tune the LLM to enhance its performance on the example of the target task. You will accomplish this by scoring the model’s performance on the target task after being fine-tuned with the provided sample data.

### Instructions:

1. Consider how well the sampled training data aligns with the example in terms of:
  - Semantic similarity: How similar are the contents or instructions to the target task example?
  - Instruction format compatibility: Are the input-output structures of the selected data compatible with the target task?
  - Potential for improving generalization to this target task: How much does the sampled data appear to address challenges in the target task?
2. Rate the effectiveness of the training data on a scale of 1 to 10 for the target task example, where:
  - 1 means the training data is completely irrelevant or harmful.
  - 10 means the training data is highly relevant and likely to maximize performance.
3. Provide a short explanation for your rating.

### Representative Example:

{ }

### Sampled Training Data:

{ }

### Output:

For the given target task example and sampled training data, provide the following:

1. A rating (1-10) for the sampled data based on the criteria above.
2. A brief explanation justifying your rating.

## B.2 LLM Data Analyst Details

We select GPT-4o-mini [48] as the LLM data analyst. The prompt used is shown above. When calling the API, the temperature is set to 0.8.

### B.3 Polysemantic Neuronal Activation Extraction Details

We adopt the method from CATS [57] to extract polysemantic neuronal activations from a specified layer of the large language model, specifically within the Gated-MLP block. Mathematically, let  $\mathbf{x} \in \mathbb{R}^{d_1}$  denote the input to the Gated-MLP. The polysemantic neuron activation pattern is computed as

$$\mathbf{z}^{\text{poly}} = \text{TopK}(|\text{SiLU}(\mathbf{W}_{\text{gate}}\mathbf{x})|), \quad (10)$$

where  $\mathbf{W}_{\text{gate}}$  is the learnable weight matrix in the block, and  $\text{SiLU}(\cdot)$  is the activation function [58]. The operator  $\text{TopK}(\cdot)$  retains only the largest  $K$  values of the input vector and sets all remaining entries to zero, consistent with the extraction procedure for the monosemantic activations.

To ensure a fair comparison, both polysemantic and monosemantic activations are extracted from the same layer and with the same value of  $K$ , as used in the results of Figure 1-c and Figure 4.

## C Baseline Details

### C.1 MATES

We implement MATES based on the official repository<sup>14</sup>. Similar to the original work, we trained a data model based on `bert-base-uncased` [59] with a maximum sequence length of 4096. The batch size is 32, the learning rate is 5e-5, and we trained for 5 epochs with a weight decay of 0.01. All other hyperparameters were kept consistent with the original work.

### C.2 LESS

The experiments were run using the official repository<sup>15</sup>. Apart from the hyperparameters mentioned in § 3.1, all other parameters remain consistent with the original work.

### C.3 DLRDS

For this method, we considered two data selection models: `bge-base-en-v1.5` and `LLaMA3-8B`. In Appendix D.2, since the experiments were conducted on Chinese data, we used `bge-base-zh`. When selecting data with this method, we applied the Cosine similarity metric. When using `bge-base-en-v1.5` or `bge-base-zh`, if a sentence exceeded the model maximum token length (e.g., 512), we split the sentence into non-overlapping chunks that fit within the length limit. The embedding for the entire sentence was computed as the average of the embeddings of all chunks.

### C.4 BM25

To ensure speed, we used the `bm25s` [60] toolkit<sup>16</sup>.

### C.5 DSIR

We implement based on the official repository<sup>17</sup>.

### C.6 LLM2Vec

We used the official open-source implementation<sup>18</sup> and the released model `LLM2Vec-Sheared-LLaMA-mntp` for all experiments with this method.

## D Additional Experimental Results and Analysis

### D.1 Case Study: LLM as a Data Analyst

Figure 5 shows that, compared to other data selection methods, the LLM data analyst consistently finds that our approach has a clear advantage. To further analyze why MONA receives higher scores,

<sup>14</sup><https://github.com/cxcscmu/MATES>

<sup>15</sup><https://github.com/princeton-nlp/LESS>

<sup>16</sup><https://github.com/xhluca/bm25s>

<sup>17</sup><https://github.com/p-lambda/dsir>

<sup>18</sup><https://github.com/McGill-NLP/llm2vec?tab=readme-ov-file>

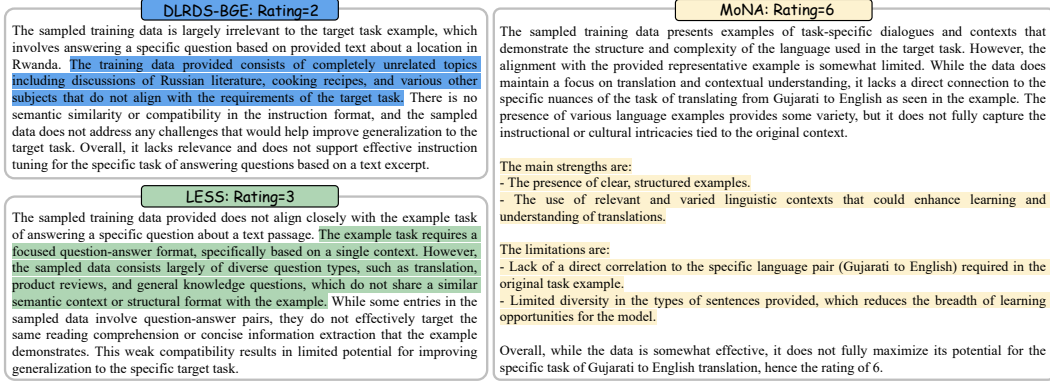


Figure 9: Explanation of the score given by the LLM data analyst for data selected by different methods

Table 4: Recall rate of the knowledge required for physical questions across different methods

Method	DLRDS-BGE	LESS	MoNA (ours)
Hit@10	78.69	13.93	81.15

we provide a case study in Figure 9, which illustrates the LLM analyst evaluation for the same TydiQA example selected by different methods. In this case, DLRDS-BGE receives a low score due to selecting irrelevant topics, while LESS is penalized for choosing data that does not match the question-answer format. These low scores are justified, since empirical evidence indicates that semantic relevance and format alignment between training and test data are critical for downstream performance. In contrast, MoNA selects data satisfying both criteria, even though the samples are in different language pairs. The LLM data analyst considers this selection acceptable, likely because semantically similar content across languages can also improve model performance, as supported by previous work [61, 62].

## D.2 Knowledge Recall for Domain-Specific Questions

To further evaluate data selection quality on specialized tasks, we conduct a knowledge recall experiment using the physics subset from [63]. Each of the 122 physics questions covers specific knowledge points, with a total of 84 distinct points represented. For each data selection method, we measure the recall rate of relevant knowledge, that is, the proportion of target knowledge points that are present in the selected training data. As reported in Table 4, MoNA achieves higher recall rates compared to DLRDS-BGE, while LESS demonstrates a low recall success rate. This difference likely arises from a greater mismatch between the distribution of knowledge data and question data for some methods, leading to larger gradient discrepancies and less effective selection. These results indicate that MoNA is more effective in retrieving domain-specific knowledge essential for answering specialized questions.

## D.3 Effect of Data Relevance on Fine-Tuning Performance

To assess the impact of data relevance, we conduct experiments comparing fine-tuning performance using data that is either highly similar or highly dissimilar to the target task, as determined by Jaccard similarity. Specifically, we select subsets of training samples that are the most similar or most dissimilar to the target task, and use each subset for instruction tuning. As shown in Table 5, using dissimilar data for fine-tuning leads to a significant reduction in performance across benchmarks, especially on GSM8K, whereas using similar (relevant) data consistently yields better results. These findings underscore the importance of selecting task-relevant data and validate the effectiveness of Jaccard similarity in our data selection framework.

Table 5: Performance comparison of fine-tuning with data dissimilar versus similar to the target task (measured by Jaccard similarity,  $\mathcal{D}^{\text{src}} = \text{OPENHERMES-2.5}$ )

Selection	MMLU	GSM8K	BBH	Avg.
dissimilar	63.38	33.06	61.17	52.54
similar	64.49	67.93	66.44	66.29

Table 6: Performance of different data curation methods on TydiQA ( $\mathcal{D}^{\text{src}} = \text{LESS}$ )

Method	Data Curation Paradigm	TydiQA
GPT-4o (Self-instruct)	data synthesis	46.99
MoNA (ours)	data selection	72.60

#### D.4 Comparison with Self-Instruct Data Synthesis

To further compare data curation paradigms, we evaluate MoNA against a GPT-4o-based self-instruct method [64]<sup>19</sup> that synthesizes instruction data (data synthesis), as commonly adopted in recent literature. Table 6 reports the TydiQA performance for both approaches. MoNA (data selection) significantly outperforms GPT-4o (self-instruct), achieving a score of 72.60 versus 46.99. This disparity can be attributed to the tendency of self-instruct methods to generate large amounts of data that may become increasingly redundant and less diverse as the volume grows. In contrast, MoNA leverages data selection to curate a more relevant and diverse subset from an existing dataset. When a sufficiently large pool of real data is available, selecting high-quality samples may be a more effective and straightforward approach than synthesizing new data.

#### D.5 Analysis of Length Bias in Token Aggregation

**Empirical Analysis** Table 7 presents the performance and average selected sample length for random selection, MoNA without length normalization, and MoNA with length normalization across multiple tasks. When sum aggregation is used (i.e., without normalization), MoNA tends to select samples whose lengths are closest to the average length of the representative target samples, rather than samples with the highest semantic relevance, and model performance suffers as a result. In contrast, applying length normalization (mean aggregation) effectively removes this bias, enabling the selection of samples with a broader range of lengths and higher semantic utility. This results in significant improvements in downstream task performance. These empirical findings demonstrate that length normalization in token aggregation is essential for mitigating length bias and achieving more reliable data selection for MoNA.

**Theoretical Analysis** Below, we show why sum aggregation induces a length bias, and how mean aggregation mitigates it. Let each sample  $\mathbf{s}_i$  consist of  $n_i$  tokens, each mapped to a non-negative  $d$ -dimensional activation vector  $\mathbf{z}_{i,j}$ . Under sum aggregation, the sample embedding is

$$\mathbf{z}_i = \sum_{j=1}^{n_i} \mathbf{z}_{i,j}. \quad (11)$$

We use the generalized Jaccard similarity between  $\mathbf{z}_i$  and the task prototype  $\mathbf{z}^{\text{tgt}}$ , defined as:

$$\mathcal{J}(\mathbf{z}_i, \mathbf{z}^{\text{tgt}}) = \frac{\sum_k \min(\mathbf{z}_i[k], \mathbf{z}^{\text{tgt}}[k])}{\sum_k \max(\mathbf{z}_i[k], \mathbf{z}^{\text{tgt}}[k])}. \quad (12)$$

In sum aggregation, the magnitude of  $\mathbf{z}_i[k]$  scales linearly with the number of tokens  $n_i$ , since  $\mathbf{z}_i[k] \approx m_{i,k} \cdot v_k$ , where  $m_{i,k}$  is the number of tokens in  $\mathbf{s}_i$  that activate dimension  $k$ , and  $v_k$  is the typical activation value. Similarly,  $\mathbf{z}^{\text{tgt}}[k]$  reflects the activation strength of the target prototype, which is influenced by the average length of representative samples of the target task. When the length  $n_i$  of  $\mathbf{s}_i$  is similar to the length associated with  $\mathbf{z}^{\text{tgt}}$ , the magnitudes of  $\mathbf{z}_i[k]$  and  $\mathbf{z}^{\text{tgt}}[k]$  are comparable. If their activated dimensions overlap well, the Jaccard similarity is maximized. However:

<sup>19</sup><https://github.com/yizhongw/self-instruct>

Table 7: Performance and average selected sample length for RANDOM, MoNA w/o length normalization, and MoNA w/ length normalization (LN) across tasks ( $\mathcal{D}^{\text{src}} = \text{OPENHERMES-2.5}$ ). The “Target Length” column shows the average length of representative target samples.

Task	Target Length	RANDOM		MoNA w/o LN		MoNA w/ LN	
		Performance	Length	Performance	Length	Performance	Length
MMLU	150.85	64.39		62.88	152.76	64.49	499.41
GSM8K	189.66	60.05		64.06	212.70	67.93	322.74
BBH	303.49	64.63	391.34	65.55	313.82	66.44	492.14
MBPP	110.38	49.67		46.20	202.67	48.40	404.03
GPQA	370.54	30.43		27.00	309.88	31.47	632.61

Table 8: Performance comparison using LESS data: polysemantic vs. monosemantic activations

Neuronal Activation	MMLU	BBH	TydiQA	Avg.
polysemantic	63.57	63.28	71.01	65.95
monosemantic	64.78	64.21	72.60	67.20
w/o SAE	62.11	61.54	71.91	65.19

- If  $n_i$  is much larger than the effective length of  $\mathbf{z}^{\text{tgt}}$ , then  $\mathbf{z}_i[k] \gg \mathbf{z}^{\text{tgt}}[k]$  for many dimensions  $k$ , leading to  $\min(\mathbf{z}_i[k], \mathbf{z}^{\text{tgt}}[k]) = \mathbf{z}^{\text{tgt}}[k]$  and  $\max(\mathbf{z}_i[k], \mathbf{z}^{\text{tgt}}[k]) = \mathbf{z}_i[k]$ . Consequently,  $\mathcal{J}(\mathbf{z}_i, \mathbf{z}^{\text{tgt}}) \approx \frac{\sum_k \mathbf{z}^{\text{tgt}}[k]}{\sum_k \mathbf{z}_i[k]} \ll 1$ , reducing the similarity.
- If  $n_i$  is much smaller,  $\mathbf{z}_i[k] \ll \mathbf{z}^{\text{tgt}}[k]$ , yielding  $\mathcal{J}(\mathbf{z}_i, \mathbf{z}^{\text{tgt}}) \approx \frac{\sum_k \mathbf{z}_i[k]}{\sum_k \mathbf{z}^{\text{tgt}}[k]} \ll 1$ , again lowering the similarity.

Thus, samples with lengths  $n_i$  close to the length of  $\mathbf{z}^{\text{tgt}}$  achieve higher similarity scores, introducing a length bias in the selection process.

In contrast, mean aggregation normalizes each sample embedding by the number of tokens. This normalization ensures that the magnitude of the embedding does not depend on the sample length. As a result, the selection process is no longer biased toward samples with lengths similar to the prototype. Instead, comparisons focus purely on the similarity of activation patterns, eliminating the systematic length bias observed with sum aggregation and enabling selection based on semantic relevance.

## D.6 Effect of SAE

Monosemantic activations, produced by the sparse autoencoder, consistently outperform polysemantic activations across all tasks on the LESS dataset (Table 8). This demonstrates that disentangling neuron polysemanticity via SAE leads to more effective data selection and superior downstream performance. Additionally, we report results using the raw hidden states of the selected layer without SAE mapping (“w/o SAE”). This baseline underperforms both polysemantic and monosemantic representations, underscoring the importance of explicit disentanglement with SAE for optimal data selection.

Table 9: Lexical Diversity (MTLD, larger is better) of Selected data

Method	MMLU	BBH	TydiQA
RANDOM	61.31	61.31	61.31
MATES	68.25	56.83	81.01
LESS	76.74	53.56	77.80
BM25	74.89	58.41	66.65
DSIR	52.10	48.90	54.01
DLRDS-BGE	50.79	44.63	70.63
DLRDS-LLaMA3-8B	49.38	40.50	82.15
MoNA (ours)	66.05	42.49	81.38

## D.7 Diversity of Selected Data by MoNA

We measure the lexical diversity of the selected data samples using the MTLT (Measure of Textual Lexical Diversity) metric [65], computed with the LexicalRichness<sup>20</sup> package. A higher MTLT score indicates greater lexical diversity. Table 9 confirm that a stronger emphasis on data quality in target-specific selection indeed leads to reduced data diversity.

## D.8 Detailed Experimental Results

We present complete results for all tasks and settings discussed in the main text. To facilitate understanding, we summarize key findings in the main text using figures and overview tables, while the following tables provide detailed, task-level results and additional experimental details. These comprehensive tables (10, 11, 12, 13, 14) complement the main text by offering the full performance breakdown, allowing readers to reproduce, verify, or further analyze various aspects of the experiments.

Table 10: Detailed results of Figure 3 ( $\mathcal{D}^{\text{src}} = \text{LESS}$ )

Method	1%				10%			
	MMLU	BBH	TyDiQA	Avg.	MMLU	BBH	TyDiQA	Avg.
RANDOM	65.71	6.77	61.77	44.75	63.23	63.35	71.14	65.90
MATES	63.22	53.00	51.15	55.79	63.62	64.28	70.04	65.98
LESS	63.58	56.59	57.31	59.16	63.27	61.77	70.56	65.20
BM25	63.96	47.50	67.36	59.61	63.47	62.06	71.15	65.56
DSIR	64.91	0.20	56.73	40.61	63.17	61.07	67.22	63.82
DLRDS-BGE	64.71	65.58	52.74	61.01	63.45	61.59	71.61	65.55
DLRDS-LLaMA3-8B	63.48	59.53	62.31	61.77	62.28	59.91	72.57	64.92
MoNA (ours)	64.57	56.67	66.77	62.67	62.85	62.94	72.74	66.18

Table 11: Detailed results of Figure 5

Method	GSM8K ( $\mathcal{D}^{\text{src}} = \text{OPENHERMES-2.5}$ )	TyDiQA ( $\mathcal{D}^{\text{src}} = \text{LESS}$ )
DLRDS-BGE	1.35	2.33
LESS	1.56	2.44
MoNA (ours)	4.39	4.11

Table 12: Detailed results of Figure 6-(a) ( $\mathcal{D}^{\text{src}} = \text{OPENHERMES-2.5}$ )

Layer	MMLU	GSM8K	BBH	MBPP	GPQA	Avg.
8	63.94	63.15	64.75	48.20	31.92	54.39
12	64.06	63.46	64.75	48.40	29.24	53.98
16	64.73	67.02	66.47	49.20	30.80	55.64
20	64.36	68.54	65.21	49.00	30.80	55.58
24	64.13	69.52	65.27	52.00	27.68	55.72
26	63.11	67.10	64.06	49.20	31.03	54.90
31	64.49	67.93	66.44	48.40	31.47	55.75

Table 13: Detailed results of Figure 6-b ( $\mathcal{D}^{\text{src}} = \text{OPENHERMES-2.5}$ )

$K$	192	96	48	24
GSM8K	67.93	67.48	66.41	64.29

## E Algorithm

A complete description of our algorithm is provided (Algorithm 1) in the form of pseudocode, to facilitate reproducibility and implementation in future work.

<sup>20</sup><https://github.com/LSYS/LexicalRichness>

Table 14: Detailed results of Figure 6-c ( $\mathcal{D}^{\text{src}} = \text{OPENHERMES-2.5}$ )

<b>Method</b>	<i>MMLU</i>	<i>GSM8K</i>	<i>BBH</i>	<b>Avg.</b>
Cosine	63.30	66.72	61.33	63.78
Euclidean	63.03	65.73	61.62	63.46
Jaccard	64.49	67.93	66.44	66.29

---

**Algorithm 1** MoNA: Task-Specific Data Selection with Monosemantic Neuronal Activations

---

**Require:** Source dataset  $\mathcal{D}^{\text{src}}$ , target set  $\mathcal{D}^{\text{tgt}}$ , data selection model  $\mathcal{M}^{\text{ds}}$ , chosen layer  $L$ , trained SAE, sparsity  $K$ , selection size  $n$

**Ensure:** Selected subset  $\mathcal{D}^{\text{sel}} \subset \mathcal{D}^{\text{src}}$  of size  $n$

- 1: **For** each sample  $\mathbf{s}$  in  $\mathcal{D}^{\text{src}} \cup \mathcal{D}^{\text{tgt}}$ :
  - 2:   Compute monosemantic activation  $\mathbf{z}_s$  as in Eq. (6) and aggregate to sample-level as in Eq. (7)
  - 3:   Compute target prototype  $\mathbf{z}^{\text{tgt}}$  using Eq. (8) on all  $\mathbf{z}_j$  in  $\mathcal{D}^{\text{tgt}}$
  - 4:   **for** each source sample  $\mathbf{s}_i$  in  $\mathcal{D}^{\text{src}}$  **do**
  - 5:     Compute similarity  $s_i$  between  $\mathbf{z}_i$  and  $\mathbf{z}^{\text{tgt}}$  as in Eq. (9)
  - 6:   **end for**
  - 7:   Select  $\mathcal{D}^{\text{sel}} =$  the  $n$  samples in  $\mathcal{D}^{\text{src}}$  with highest similarity  $s_i$
  - 8: **return**  $\mathcal{D}^{\text{sel}}$
-