TRACE: Trajectory-based Activation Change Estimation for Task-specific Data Selection

Ye He[1]*, Shangzhan Li[1], Yuxin Zhou[1], Qi Shi[2]

[1]Harbin Institute of Technology, Harbin, China
[2]Tsinghua University, Beijing, China
{yehe, szli, yxzhou}@ir.hit.edu.cn, qshi9510@gmail.com

Abstract

Task-specific data selection, which aims to identify the most relevant training instances from a large corpus to optimize performance on a target task, is a critical challenge in modern AI. Prevailing methods typically rely on either representation clustering or gradient-based influence estimation. However, these approaches have notable limitations. Representation-based methods rely on static features; they measure semantic proximity but are agnostic to the process of learning. Conversely, influence-based methods, while capturing optimization directions, often focus narrowly on aligning with the validation loss, which may not fully correlate with the desired capabilities. To address these issues, we propose TRACE, a novel algorithm that simultaneously considers data consistency in the optimization direction and representation space, and performs **TR**ajectory-based **A**ctivation **C**hange **E**stimation to select instruction data. Specifically, TRACE first performs a targeted weight update using the validation set. It then captures the optimization trajectory by calculating the change in neuron activations for each before and after this update. By selecting data whose activation change are most similar to those of the validation set, TRACE ensures alignment in both the representational and optimization domains. Our experiments demonstrate that TRACE outperforms baseline methods across various tasks, particularly in complex, data-scarce scenarios.

Code — https://github.com/godlikehhd/TRACE

Introduction

The central challenge in the current era of instruction tuning (Ouyang et al. 2022) for Large Language Models (LLMs) has shifted from generating data to curating it. While vast datasets now exist (Ding et al. 2023; Longpre et al. 2023), it is now well-established that data quality, not quantity, determines model performance. This principle becomes especially critical for targeted applications, where the challenge of selecting the most relevant data to optimize a model for a specific downstream task remains a significant open problem. The central challenge in task-specific data selection is that the objective—maximizing final task performance—is computationally intractable. All practical methods must therefore optimize for a tractable proxy metric, but a fundamental mismatch often exists between these proxies and the ultimate goal.

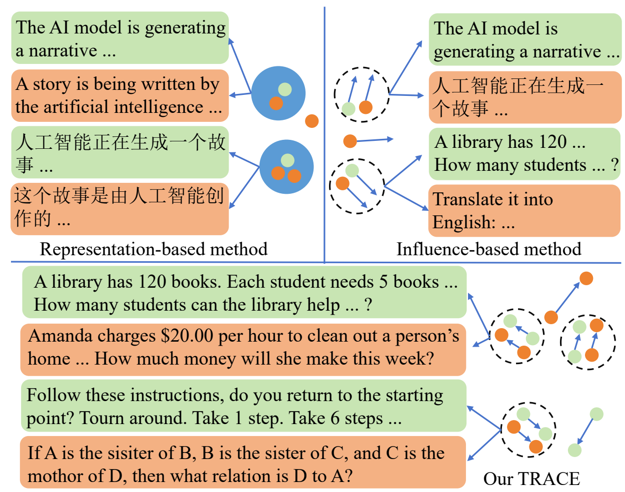

Figure 1: Comparison between different methods.

Existing methods predominantly fall into two paradigms. As shown in Figure 1. Representation-based methods use a distributional proxy, which select data by aligning distributions like n-gram (Xie et al. 2023) or embeddings (Zhang et al. 2018). All those methods compute the distance of data under certain semantic space, and the data selected under such conditions will reflect the properties of that space (e.g., data in the same language are more similar, etc.). While interpretable, relying on representation similarity is suboptimal as it fails to account for the model's optimizing direction. The second category uses an optimization proxy, which attempts to capture optimizing dynamics by estimating a training instance's influence, often through gradient alignment (Xia et al. 2024). These methods are built on the principle that reducing validation loss is highly correlated with improving task performance. Although it's more direct, studies (Zhou et al. 2023) reveal a tenuous correlation between loss reduction and model's performance, particularly for tasks with complex evaluation metrics.

Given these limitations, we propose that a more faithful proxy for data selection is the consistency of representa-

*Corresponding Author.

tional dynamics under task-specific optimization. The key insight is that the vector of change in a data point's representation synthesizes two crucial elements: the "force" of the optimization direction and the "landscape" of the semantic space. Motivated by the established link between a model's internal activation patterns and its task-specific behaviors (Wang et al. 2022), we use the neuron activation states as the features and select relevant data. Recently, MONA (Ma et al. 2025) utilizes sparse autoencoder to encode neuronal activations and select data with similar representations. Although it consider the property of neuron, it utilizes representation similarity for data selection, which still ignore the optimization target.

To alleviate the above limitations, we introduce Trajectory-based Activation Change Estimation (TRACE), a method that unifies distributional and optimization-aware selection. TRACE begins by performing a one-step optimization on the target validation set, which induces a measurable "activation change" (see Figure 1). We conceptualize this change embodies the model's optimization trajectory. Based on the hypothesis that relevant training data should produce an analogous change, we select data by measuring the similarity of these activation changes between the training and validation sets.

We validate TRACE through extensive experiments, demonstrating its consistent superiority over existing data selection baselines. The method's efficacy is particularly highlighted on the multilingual TyDiQA benchmark, where it achieves an average absolute improvement of 2 performance points over full-dataset fine-tuning across three models, confirming its ability to select a highly potent, task-aligned data subset. In addition, we further provide a theoretical analysis of TRACE's mechanism, along with a discussion of its limitations and future research directions.

Preliminaries

In this section, we first provide a background on influence-based methods. Subsequently, we introduce TRACE, and elucidate how it advances beyond existing methods by jointly considering both representational features and optimization dynamics.

Influence Estimation via Gradient Alignment

Influence-based approaches aim to estimate the direct impact of a training instance on the model's performance on a validation set. This is typically achieved by leveraging a first-order Taylor expansion to approximate the training dynamics.

Formally, consider a model π_θ^t with parameters θ at training step t. After a single optimization step using a training instance z, the parameters are updated to θ^{t+1}. The change in the model's loss on a validation datapoint z', denoted as $l(z';\theta)$, can be approximated by its first-order Taylor expansion:

$$l(z';\theta^{t+1}) - l(z';\theta^t) \approx \langle \nabla_\theta l(z';\theta^t),\ \theta^{t+1} - \theta^t \rangle \tag{1}$$

the parameter update is simply the negative gradient of the training loss,$\theta^{t+1} - \theta^t = -\eta_t \nabla_\theta l(z;\theta^t)$. Substituting this in gives the classic influence formula:

$$l(z';\theta^{t+1}) - l(z';\theta^t) \approx -\eta_t \langle \nabla_\theta l(z';\theta^t),\ \nabla_\theta l(z;\theta^t) \rangle \tag{2}$$

It is easily understood that when the gradient of the validation set data is in the same direction as the gradient of the training set data, we have $l(z';\theta^{t+1}) - l(z';\theta^t) < 0$, that is, the higher the gradient similarity between the training data and the validation data, the more the validation set loss is reduced. Building on this, (Pruthi et al. 2020) models the influence of training data by aggregating their effects across multiple epochs during training.

$$\text{Inf}(z, z') \triangleq \sum_{i=1}^{N} \bar{\eta}_i \langle \nabla \ell(z';\boldsymbol{\theta}_i), \nabla \ell(z;\boldsymbol{\theta}_i) \rangle \tag{3}$$

Tajectory-based Activation Change Estimation

In this section, we establish the theoretical underpinnings of TRACE. We first formalize the process of task-specific fine-tuning and then derive an analytical expression for our core metric, the activation change, revealing its connection to influence-based methods and its ability to jointly capture semantic and optimization alignment.

We begin by considering an initial base model, π_{base}, with parameters θ_{base}. We perform a single-step fine-tuning update on this model using the validation set, D_{val}, to obtain an updated model, π_{val}, with parameters θ_{val}. Based on first-order Taylor expansion, this parameter update can be approximated as:

$$\Delta\theta = \theta_{\text{val}} - \theta_{\text{base}} \approx -\eta \nabla_\theta L_{\text{val}}(\theta_{\text{base}}) \tag{4}$$

where $L_{val}(\theta_{base})$ is the loss on the validation set, and η is the learning rate. The gradient term, $\nabla_\theta L_{\text{val}}(\theta_{\text{base}})$, represents the direction in the parameter space that yields the steepest descent on the validation loss, thus encapsulating the desired optimization direction.

The core of our method is to analyze the representational shift this parameter update induces on the neuron activations. We define this shift as the activation change, $\delta_h(x)$, for any given input x:

$$\delta_h(x) = h(x;\theta_{val}) - h(x;\theta_{base}) \tag{5}$$

Here, $h(x;\theta)$ denotes the neuron activation vector from a specific layer in the model. To understand the intrinsic meaning of $\delta_h(x)$, we approximate $h(x;\theta_{val})$ by applying a first-order Taylor expansion to $h(x;\theta)$ around the point θ_{base}:

$$h(x;\theta_{val}) \approx h(x;\theta_{base}) + (\theta_{val} - \theta_{base})^T \cdot \nabla_\theta h(x;\theta_{base}) \tag{6}$$

where $\nabla_\theta h(x;\theta_{base})$ is the Jacobian of the activation function with respect to the model parameters. By substituting the parameter update approximation into this expansion, we arrive at an analytical form for the activation change:

$$\delta_h(x) \approx -\eta \left(\nabla_\theta L_{val}(\theta_{base})\right)^T \cdot \nabla_\theta h(x;\theta_{base}) \tag{7}$$

This formulation reveals the essence of the activation change: it is the projection of the model's activation sensitivity for input x(the Jacobian term) onto the validation

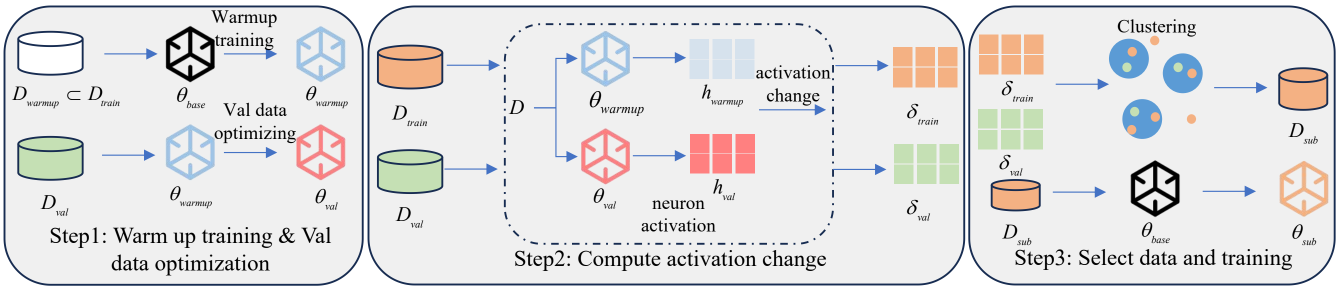

Figure 2: Illustration of TRACE. First we perform warmup training using $\mathcal{D}_{warmup}$ and obtain θ_{warmup}. Then we use $\mathcal{D}_{val}$ to fine-tune θ_{warmup} to obtain θ_{val}. Second, we compute activation change δ_{train} and δ_{val} of the Neuron Activation of $\mathcal{D}_{train}$ and $\mathcal{D}_{val}$. Finally, we select data $\mathcal{D}_{sub}$ by compute the TRACE score between δ_{train} and δ_{val} and then train a task-specific model.

task's optimal update direction (the gradient term). In other words, $\delta_h(x)$ not only contains the semantic information of x(embedded within $h(x)$ and its Jacobian) but, more importantly, it quantifies the direction and magnitude of change in its internal representation as the model evolves toward the validation objective.

Based on this derivation, TRACE selects data by measuring the alignment of these representational shifts. For each training sample $x_i \in D_{train}$, we compute its activation change $\delta_h(x_i)$ and compare it to the vector representing the shift of the validation set, $\delta_h(x_j \in D_{val})$, via cosine similarity:

$$\text{score}(x_i, x_j) = \cos(\delta_h(x_i), \delta_h(x_j)) \tag{8}$$

A high score signifies that the training sample x_i induces a representational shift that is highly aligned with that of the validation data, making it a valuable candidate for targeted fine-tuning.

Method

In this section, we first define the task-specific data selection problem and its objective. Subsequently, we present an overview of TRACE, followed by a detailed exposition of its implementation.

Task Definition

The problem of Task-Specific Data Selection (TSDS) is to identify a subset of data from a large-scale training corpus. Formally, given a large training set $\mathcal{D}_{train}$, a small, task-specific validation set $\mathcal{D}_{val}$ which serves as a guide, and a final test set $\mathcal{D}_{test}$, the goal is to select a subset $\mathcal{D}_{sub} \subset \mathcal{D}_{train}$ of a predetermined size k. The objective is to find the optimal subset D^*_{sub} such that a model trained on it, $\pi_{\theta(D^*_{sub})}$, maximizes a performance metric S on the unseen test set $\mathcal{D}_{test}$.

This objective can be expressed as finding the subset that solves the following optimization problem:

$$\mathcal{D}^*_{sub} = \mathop{\arg\max}_{\mathcal{D}_{sub} \subset \mathcal{D}_{train}, |\mathcal{D}_{sub}|=k} \mathcal{S}(\pi_\theta(\mathcal{D}_{sub}), \mathcal{D}_{test}) \tag{9}$$

where $S(\cdot,\cdot)$ is an evaluation metric, such as accuracy or F1-score, measuring the model's performance on the specified task.

Implementation Details

As illustrated in Figure 2, TRACE is a multi-stage process designed to select task-relevant data from a large pool. The process begins with a warmup training phase to adapt the model, followed by a task-specific fine-tuning step to capture the optimization direction. Subsequently, we extract and compare activation change to score and select training instances.

Warmup Training Consistent with findings in prior work (Xia et al. 2024), we empirically observe that directly fine-tuning a base pre-trained model on the validation set yields a suboptimal signal for data selection. To mitigate this, we first perform a warmup stage. We train a base model π_{base} for one epoch on a small subset of the candidate data pool, $\mathcal{D}_{warmup} \subset \mathcal{D}_{train}$, to produce an initial instruction-aligned model, π_{warmup}. The data for this stage, $\mathcal{D}_{warmup}$, is selected either by random sampling or through k-means clustering on semantic features to ensure diversity (see Ablation for details).

Following the warmup, we perform a single optimization step. The entire validation set, $\mathcal{D}_{val}$, is processed in a single batch to fine-tune π_{warmup}. This yields the validation-aligned model, π_{val}, which encapsulates the desired learning trajectory for the target task.

Activation Change Estimation Our selection criterion is based on the representational shifts within the model's intermediate layers. For a given Transformer block in an LLM, the feed-forward network (FFN) operation is typically:

$$FFN(z) = (SiLU(zW_{gate}) \odot zW_{up})W_{down} \tag{10}$$

Following prior work (Wang et al. 2022; Dai et al. 2022; Gurnee et al. 2023), we select the output of the activation function as the neuron activation feature for a given layer:

$$h(z) = SiLU(zW_{gate}) \tag{11}$$

Algorithm 1: Select Training Samples Based on Similarity

Require: Similarity matrix $\texttt{sims} \in \mathbb{R}^{N\times M}$, target number $\texttt{num}$, number of categories $\texttt{type\_num}$, category size $\texttt{cato\_size}$
Ensure: Selected training sample indices $\texttt{idx}$ of length $\texttt{num}$

$\texttt{idx} \leftarrow \emptyset$
$\texttt{num\_cur} \leftarrow \lfloor \texttt{num}/\texttt{type\_num} \rfloor$
for $i = 0$ to $\texttt{type\_num} - 1$ **do**
  $\texttt{slice} \leftarrow \texttt{sims}[:, i \cdot \texttt{cato\_size} : (i+1) \cdot \texttt{cato\_size}]$
  $\texttt{mean\_sim} \leftarrow$ mean of each row in $\texttt{slice}$
  $\texttt{top\_indices} \leftarrow$ indices of top $\texttt{num\_cur}$ entries in $\texttt{mean\_sim}$
  Add $\texttt{top\_indices}$ to $\texttt{idx}$
end for
return $\texttt{idx}$

To create a holistic representation for each data instance $z = \{x, y\}$, where x is the prompt and y is the output, we concatenate them and perform a single forward pass. Using forward hooks, we extract the activation matrix $h(z) \in \mathbb{R}^{L\times d}$ for a specific layer, where L is the sequence length of the concatenated input and d is the hidden dimension. To handle variable sequence lengths and obtain a fixed-size vector representation, we apply mean-pooling across the sequence length dimension:

$$ac(z) = \frac{1}{L}\sum_{i=1}^{L} h(z)_i \tag{12}$$

We then compute this vector representation, $ac(z)$, for all training and validation data using both the π_{warmup} and π_{val} models.

Next, we calculate the activation difference for each training and validation instance:

$$\delta_{ac}(z) = ac^{val}(z) - ac^{warmup}(z) \tag{13}$$

Finally, we score each training instance z_i based on the cosine similarity between its activation change and the activation change of the entire validation set.

$$\text{score}(z_{train}, z_{val}) = \cos(\delta_{ac}(z_{train}), \delta_{ac}(z_{val})) \tag{14}$$

Data Selection For a given target task, we begin by computing the pairwise Neuron Activation Change Similarity scores between every training instance in $\mathcal{D}_{train}$ and every validation instance in $\mathcal{D}_{val}$. This procedure yields a score matrix $\mathbf{S} \in \mathbb{R}^{N\times M}$, where $N = |\mathcal{D}_{train}|$ and $M = |\mathcal{D}_{val}|$. We observe that validation sets are typically heterogeneous, comprising multiple distinct sub-tasks that may differ in format and semantics. To ensure our selection process is sensitive to this structure, we adopt the sub-task-aware strategy outlined in algorithm 1. Finally, we can obtain the subset D_{sub} for specific task.

Experiments

Experiment Setup

Training Data and Evaluation Task To ensure a rigorous and comparable evaluation, we adopt the experimental setup established by prominent prior work in targeted data selection. Our source data pool is a composite of four widely-used instruction-tuning datasets—Flan v2 (Longpre et al. 2023), CoT (Wei et al. 2022), Dolly (Mike et al. 2023), and Open Assistant (Köpf et al. 2023), totaling approximately 270,000 examples. For evaluation, follow LESS (Xia et al. 2024) and MONA (Ma et al. 2025), we utilize the following benchmark for evaluation: MMLU (Hendrycks et al. 2021) (multi-domain knowledge), BBH (Srivastava, Rastogi, and et al 2023) (complex reasoning), and TyDiQA (Clark et al. 2020) (multilingual QA). Detailed information of these dataset is available in Appendix.

Training Details We evaluate TRACE method on three Large Language Models: LLaMA-3.1-8B (Dubey et al. 2024), OLMo-7B (Groeneveld et al. 2024), and Qwen-2.5-7B (Team 2024). We investigate two strategies for constructing the warmup dataset: (1) Random Sampling, where a subset is selected based on three random seed, we report the average score to ensure the reliability of TRACE, and (2) Clustering-based Sampling, where k-means clustering is applied to semantic features to ensure representative diversity. We trained warmup data for 1 epoch and perform TRACE to select data. In training stage, we utilize the same setting as (Ma et al. 2025) for fair comparison. All the methods select 5% data to train a model with 2 epochs and batch size is 128. Learning rate is set to 7e-6 and weight decay is 0.1. And we use LLama-Factory (Zheng et al. 2024) to perform model training. All analysis are conducted by LLama-3.1-8B. Detailed information of training is available in Appendix.

Baselines

We compare TRACE with a comprehensive set of baselines, include random selection and full data training. We also compare representation-based methods like BM25 and DSIR (Xie et al. 2023), which use TF-IDF and n-gram to select similar data. We also compare RDS (Zhang et al. 2018), a dense retrieval method that computes similarity on the hidden states of last layer, for our implementation, we use the corresponding LLM to get embedding and selet data. For MoNA (Ma et al. 2025), it employs sparse autoencoder on neuron activation to capture fine-grained features for data selection. We also compare influence-based methods, in which MATES (Yu, Das, and Xiong 2024) utilizes a small model to predict the influence of training data during training stage. And LESS (Xia et al. 2024) selects data based on the gradient similarity between training and validation instances.

Main Results

Table 1 presents the main results of our TRACE method against several baselines. The results unequivocally demonstrate the superiority of our approach. Across all tested models, TRACE consistently outperforms all baseline methods, validating its effectiveness in selecting high-value data for task-specific instruction tuning.

Another key finding is that training on a 5% subset selected by TRACE often surpasses the performance of training on the full dataset. This "less is more" phenomenon, also observed in prior work (Zhang et al. 2024; Liu et al. 2024;

Methods	BBH	MMLU	TyDiQA	AVG
Qwen-2.5-7B				
Base	64.03	74.29	70.73	69.66
Random*	62.79	**73.01**	73.47	69.76
Full*	62.11	72.95	72.3	69.12
BM25*	63.71	<u>73.00</u>	72.94	69.88
RDS*	63.32	72.40	69.36	68.36
DSIR*	63.85	**73.01**	71.62	69.49
LESS*	**66.10**	72.33	71.56	70.00
TRACE(random)	64.65	72.81	<u>74.30</u>	<u>70.58</u>
TRACE(kmeans)	<u>64.81</u>	72.3	**74.91**	**70.67**
LLama-3.1-8B				
Base	62.65	65.33	69.99	65.99
Random*	62.79	62.03	70.88	65.17
Full*	63.39	61.86	73.68	66.31
BM25*	63.71	63.03	71.76	66.17
DSIR	63.19	<u>64.25</u>	65.61	64.35
RDS*	<u>64.68</u>	62.25	71.39	66.11
LESS	62.11	62.51	70.68	65.10
MATES	63.68	63.62	67.74	65.01
MONA	64.21	**64.78**	72.60	67.20
TRACE(random)	63.69	63.17	<u>75.41</u>	<u>67.42</u>
TRACE(kmeans)	**64.80**	63.92	**76.50**	**68.41**
Olmo-7B				
Base	30.20	28.36	33.35	30.64
Random*	29.93	27.18	29.17	28.76
Full*	30.38	**42.46**	31.96	<u>34.93</u>
BM25*	27.89	29.52	25.14	27.52
DSIR	**32.87**	29.54	<u>33.25</u>	31.89
RDS*	14.41	28.74	29.42	24.19
LESS	30.07	37.21	33.20	33.49
MATES	30.46	29.57	31.02	30.35
MONA	30.19	40.14	**33.80**	34.71
TRACE(random)	31.14	<u>41.92</u>	31.00	34.69
TRACE(kmeans)	<u>31.25</u>	41.63	32.42	**35.10**

Table 1: Performance of different methods. "$*$" denotes our implementation. We report the result from (Ma et al. 2025).

Li et al. 2023), underscores the importance of data selection. We attribute this to the fact that large, heterogeneous datasets often contain noisy, irrelevant, or even conflicting examples that can hinder the acquisition of a specific capability. By calculating the activation trajectory changes of the data during model training, TRACE can select data that is more relevant to the task, and thus achieve better results than the all-data training as well as baseline methods.

We also note model-specific behaviors. On the Qwen-2.5-7B model, even random selection outperforms full-dataset training. We hypothesize that highly capable base models like Qwen may only require a small amount of targeted data to elicit a specific skill, and training on the full, lower-quality dataset can be counterproductive. Furthermore, TRACE demonstrates a particularly pronounced performance gain on the TyDiQA task with the LLaMA-3.1-8B model. By comparing data screened by different methods (see Qualitative Analysis), it reveals that TRACE successfully abstracts away from surface-level features.

Methods	BBH	MMLU	TyDiQA	AVG
Layer 11				
TRACE	63.51	60.49	76.94	66.98
TRACE(Instruct)	63.89	62.31	72.89	66.36
w/o warmup	62.49	63.59	69.97	65.35
w/o val tuning	63.05	61.83	76.59	**67.16**
Layer 12				
TRACE	62.82	63.2	76.5	**67.51**
TRACE(Instruct)	64.34	62.79	72.54	66.56
w/o warmup	64.23	62.46	69.85	65.51
w/o val tuning	63.69	62.68	74.48	66.95
Layer 13				
TRACE	63	63.5	75.75	**67.42**
TRACE(Instruct)	62.49	61.68	72.50	65.56
w/o warmup	62.63	62.11	68.2	64.31
w/o val tuning	63.45	63.05	70.68	65.73
Layer 14				
TRACE	62.76	63.18	76.33	**67.42**
TRACE(Instruct)	61.96	62.83	72.84	65.88
w/o warmup	62.99	63.8	68.37	65.05
w/o val tuning	63.32	61.38	72.69	65.80
Layer 15				
TRACE	64.8	63.92	76.5	**68.41**
TRACE(Instruct)	64.2	63.49	71.56	66.42
w/o warmup	61.36	63.49	72.93	65.93
w/o val tuning	65.32	60.75	71.37	65.81

Table 2: Comparison of TRACE Components using LLama-3.1-8B. Instruct means that we use official instruct model for data selection.

Ablation Study

To dissect the contribution of each key component within the TRACE framework, we conducted a series of ablation studies across several representative intermediate layers. The results, presented in Table 2. We conclude as follow.

The Critical Role of the Warmup Stage. We first evaluate a variant of TRACE where the initial warmup stage is omitted (w/o warmup). As the results consistently show, this configuration leads to a substantial drop in performance across all tested layers. For instance, at Layer 15, removing the warmup stage reduces the average performance by 2.48 points (from 68.41 to 65.93). This significant degradation confirms our hypothesis that the warmup phase is crucial for adapting the base model to the instruction-following distribution. Without it, the fine-tuning signal from the small validation set is insufficient to bridge the distributional gap, resulting in a noisy and ineffective signal for data selection.

The Necessity of Validation-Guided Tuning. Next, we investigate a scenario where the targeted guidance from the validation set is removed (w/o val tuning), and the activation change is instead computed between two checkpoints from the warmup phase. The results indicate that this variant is outperformed by TRACE. The performance degradation, while significant, is generally less severe than removing the

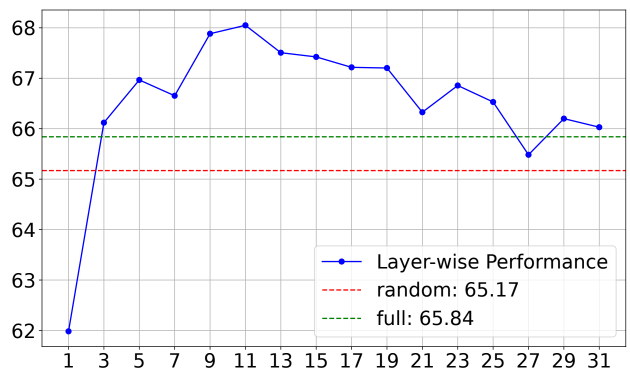

Figure 3: Layer Effectiveness on LLama-3.1-8B.

warmup stage. This suggests that the general capabilities acquired during the warmup phase (e.g., instruction following) have some positive transfer. However, the superior performance of TRACE underscores the importance of the validation set in steering the model's representational shift towards the target capability.

Layer Effectiveness

A key consideration for TRACE is the choice of which layer's activations to use. While the optimal layer may vary across different model architectures, making a priori selection non-trivial question we leave for future work. Our experiments reveal a consistent and insightful pattern. As shown in Figure 3, the performance of TRACE, generally surpasses the random selection, indicating the overall robustness. More notably, we observe that the method's efficacy peaks within the intermediate layers of the network , where TRACE consistently outperforms baseline methods. This phenomenon is not model-specific and holds true for Qwen-2.5-7B and OLMo-7B as well (see Appendix for details). This empirical result aligns with (Jawahar, Sagot, and Seddah 2019) and (Skean et al. 2025), which the former indicates that the intermediate layers of language models primarily encode abstract linguistic features, while the latter demonstrates that utilizing representations from these intermediate layers can lead to improved performance on downstream tasks.

Analysis of Multi-Checkpoint Aggregation

To assess the temporal evolution of TRACE, we conducted an experiment aggregating TRACE scores over four checkpoints from an extended warmup training phase. As shown in Table 3. Contrary to influence methods, this multi-checkpoint approach did not yield significant performance gains over single-checkpoint method. We conclude that the TRACE signal, which measures the relative representational shift induced by validation-set tuning, stabilizes remarkably quickly. The first warmup epoch appears sufficient to condition the model to the instruction-following manifold, after which its response to the targeted fine-tuning remains directionally consistent. This is strongly supported by the high overlap observed between the data subsets selected by both

	BBH	MMLU	TyDiQA	AVG
Random	62.79	62.03	70.88	65.17
1 epoch	63.38	62.92	76.60	67.63
2 epochs	64.10	62.66	75.95	67.57
3 epochs	63.02	62.25	76.20	67.15
4 epochs	63.43	62.60	76.16	67.40
overlap(%,4 epochs)	67.07	60.69	64.66	64.14

Table 3: Performance of Multi-Checkpoint Aggregation using LLama-3.1-8B. We use 11-15 layers for TRACE data selection and compute the average performance and overlap rate across 5 layers.

the single- and multi-checkpoint strategies. This finding underscores a key practical advantage of TRACE: its ability to derive a robust and effective selection signal from a computationally efficient, single-epoch warmup.

Qualitative Analysis

We conducted a qualitative analysis of the data selected for TyDiQA in Table 4. Our analysis reveals a telling dichotomy between representation-based methods like RDS and TRACE. When selecting data based on a Russian QA data, RDS selects a Russian math riddle. It correctly identifies the language but completely misses the QA task format. In contrast, TRACE selects a Spanish QA example, successfully ignoring the superficial linguistic cues to identify the representation which is beneficial for target task.

This contrast is further amplified when selecting based on all validation data (which spans nine languages). RDS, attempting to find a commonality in the diverse linguistic representations, defaults to selecting a generic multilingual task—Arabic-to-English translation. It identifies "multilingualism" as the key feature. Again, TRACE succeeds where RDS fails. It abstracts away from the varied languages and identifies the common procedural essence, selecting a high-quality, in-English question-answering example. This behavior is similar to that reported in LESS, provides strong evidence that TRACE can capture the model's optimization trajectory, allowing it to select data based on true procedural relevance. Besides, we conduct more experiments and compare more examples in the appendix section.

Related Work

Task-specific Data Selection

Instruction tuning are typically curated with one of two goals: to instill a specialized skill (e.g., code, math) (Wei et al. 2024; Hugging Face 2025; Wang et al. 2024), or to foster general instruction-following ability. (Li et al. 2024a; Chen et al. 2023; Liu et al. 2023a; Cao et al. 2023; Zhou et al. 2024; Zhang et al. 2024; Liu et al. 2024; Li et al. 2023). We explore how to effectively select a subset from a data pool to maximize performance on a specific task. It can be divided into two main paradigms: representation-based and influence-based methods.

Representation-based methods operate by aligning feature distributions between data pool and target validation set.

Validation Example of TyDiQA(Russian question answer task based on given context.)	
User: Passage: Lobachevsky taught at the Imperial Kazan University for 40 years, including 19 years as its rector; his activity and skillful leadership brought the university to the forefront of Russian educational institutions. According to N.P. Zagoskin, Lobachevsky was the "great builder" of Kazan University [2]. Question: At which university did Nikolai Ivanovich Lobachevsky teach for 40 years? **Assistant:** Imperial Kazan	
RDS(Single data, it selects a Russian math data)	TRACE(Single data, it selects a Spanish context-based QA data)
User: Kondrat went to Leningrad, And met twelve kids along the way. Each had three baskets, In each basket — a cat, Each cat had twelve kittens. Each kitten had four mice in its teeth. And Kondrat wondered: "How many mice and kittens are the kids carrying to Leningrad?" **Assistant:** Interesting story. Did you just want to share it, or do you have a question or task related to it?	**User:** Given a paragraph and a question in Spanish, answer the question from the text. Passage: [Spanish context] Question: Approximately, how many works of art are included in the Far East collections? **Assistant:** More than 70,000 works of art..
RDS(All data, it selects an Arabic translation data)	TRACE(All data, it selects an English context-based QA data)
User: You are given a sentence in Arabic. Your job is to translate the Arabic sentence into English. [Arabic Sentence] **Assistant:** During the first three years of life, the brain grows to its full size, with most of that growth taking place in the first six months. The brain develops in response to experience and to stimulation.	**User:** Student A: ESPN Monday Night Football is a live television broadcast of weekly National Football League (NFL) games on ESPN in the United States. From 1970 to 2005, it aired on sister broadcast network ABC. [More context] Question: Based on this passage, when was the last time Monday night football was on abc? **Assistant:** From 1970 to 2005, it aired on sister broadcast network ABC. 2005.

Table 4: Selected Data Comparison using LLama-3.1-8B. We compare two scenarios. "Single data": data was selected using target validation data, "All data": data was selected by averaging scores across all validation data. All data are selected with highest score. We translate them into English for easier reading.

These features can be lexical, such as n-grams, or semantic. Dubey et al. (2024) calculates the similarity of sentence features obtained by pooling the last-layer representations of a model. More recently, DSIR (Xie et al. 2023) apply n-gram to encode data representations for comparison. While interpretable, their reliance on features is a key limitation; a high feature similarity does not guarantee that data will contribute to the fine-tuning for specific task.

Influence-based methods take a more direct approach by estimating the impact of training data (Killamsetty et al. 2021; Feldman and Zhang 2020; Mirzasoleiman, Bilmes, and Leskovec 2020; Xia et al. 2020; Kaushal et al. 2019; Toneva et al. 2018). Li et al. (2024b) proposes using the change in validation loss when a training example is included in the context. Xia et al. (2024) directly measures the gradient alignment between training and validation examples. While these methods capture data influence, they ignore the fact that loss reduction is not always related to performance gain, especially in complex tasks (Zhou et al. 2023).

Neuron Activation

A parallel line of research has demonstrated that a model's internal neuron activations are highly correlated with its capabilities and behaviors (Wang et al. 2022). Different inputs can trigger distinct activation patterns corresponding to specific functions like language identification (Zhao et al. 2024; Zhu et al. 2024), format adherence (Rai and Yao 2024), or safety compliance (Zhao et al. 2025). For instance, Liu et al. (2023b) utilizes Neuron activation coverage to identify out-of-distribution data. Chen et al. (2024) explored the differences in neuron activations to locate safety-related neurons. Further, Ma et al. (2025) leveraged sparse autoencoder on neuron representations to obtain fine-grained semantic information for data selection.

Our work is situated at the intersection of these two fields. We build on the insight that neuron activations can represent specific task. However, acknowledging the limitations of representation-based methods, we draw inspiration from influence-based methods to account for the dynamic process of optimization. We therefore propose to measure the change in activations pre- and post-fine-tuning, creating a novel signal that captures the dynamics of representational geometry.

Conclusion

In this work, we introduce Trajectory-based Activation Change Estimation (TRACE), a novel method for task-specific data selection. TRACE leverages the resulting shift in neuron activations at a specific model layer as a signature for identifying valuable data. Our experimental results demonstrate the efficacy of TRACE, showing that it consistently outperforms strong baselines. The method's key advantage is particularly salient in challenging scenarios with high surface-feature diversity but a consistent underlying task structure, as exemplified by TyDiQA. Furthermore, our studies confirm the stability and robustness of the core mechanism, while also highlighting its current limitations, layer selection and scalability, which we will keep researching in the future.

Acknowledgments

We gratefully acknowledge the support of the National Natural Science Foundation of China (NSFC) via grant 62236004, 62206078 and 62476073.

References

Cao, Y.; Kang, Y.; Wang, C.; and Sun, L. 2023. Instruction mining: Instruction data selection for tuning large language models. *arXiv preprint arXiv:2307.06290*.

Chen, J.; Wang, X.; Yao, Z.; Bai, Y.; Hou, L.; and Li, J. 2024. Finding safety neurons in large language models. *arXiv preprint arXiv:2406.14144*.

Chen, L.; Li, S.; Yan, J.; Wang, H.; Gunaratna, K.; Yadav, V.; Tang, Z.; Srinivasan, V.; Zhou, T.; Huang, H.; et al. 2023. Alpagasus: Training a better alpaca with fewer data. *arXiv preprint arXiv:2307.08701*.

Clark, J. H.; Choi, E.; Collins, M.; Garrette, D.; Kwiatkowski, T.; Nikolaev, V.; and Palomaki, J. 2020. TyDi QA: A Benchmark for Information-Seeking Question Answering in Typologically Diverse Languages. *Transactions of the Association for Computational Linguistics*, 8: 454–470.

Dai, D.; Dong, L.; Hao, Y.; Sui, Z.; Chang, B.; and Wei, F. 2022. Knowledge Neurons in Pretrained Transformers. In Muresan, S.; Nakov, P.; and Villavicencio, A., eds., *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 8493–8502. Dublin, Ireland: Association for Computational Linguistics.

Ding, N.; Chen, Y.; Xu, B.; Qin, Y.; Hu, S.; Liu, Z.; Sun, M.; and Zhou, B. 2023. Enhancing Chat Language Models by Scaling High-quality Instructional Conversations. In Bouamor, H.; Pino, J.; and Bali, K., eds., *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 3029–3051. Singapore: Association for Computational Linguistics.

Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Yang, A.; Fan, A.; et al. 2024. The llama 3 herd of models. *arXiv e-prints*, arXiv–2407.

Feldman, V.; and Zhang, C. 2020. What neural networks memorize and why: Discovering the long tail via influence estimation. *Advances in Neural Information Processing Systems*, 33: 2881–2891.

Groeneveld, D.; Beltagy, I.; Walsh, E.; and Bhagia, A. e. a. 2024. OLMo: Accelerating the Science of Language Models. In Ku, L.-W.; Martins, A.; and Srikumar, V., eds., *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 15789–15809. Bangkok, Thailand: Association for Computational Linguistics.

Gurnee, W.; Nanda, N.; Pauly, M.; Harvey, K.; Troitskii, D.; and Bertsimas, D. 2023. Finding neurons in a haystack: Case studies with sparse probing. *arXiv preprint arXiv:2305.01610*.

Hendrycks, D.; Burns, C.; Basart, S.; Zou, A.; Mazeika, M.; Song, D.; and Steinhardt, J. 2021. Measuring Massive Multitask Language Understanding. In *International Conference on Learning Representations*.

Hugging Face. 2025. Open R1: A fully open reproduction of DeepSeek-R1.

Jawahar, G.; Sagot, B.; and Seddah, D. 2019. What Does BERT Learn about the Structure of Language? In Korhonen, A.; Traum, D.; and Màrquez, L., eds., *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 3651–3657. Florence, Italy: Association for Computational Linguistics.

Kaushal, V.; Iyer, R.; Kothawade, S.; Mahadev, R.; Doctor, K.; and Ramakrishnan, G. 2019. Learning from less data: A unified data subset selection and active learning framework for computer vision. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 1289–1299. IEEE.

Killamsetty, K.; Durga, S.; Ramakrishnan, G.; De, A.; and Iyer, R. 2021. Grad-match: Gradient matching based data subset selection for efficient deep model training. In *International Conference on Machine Learning*, 5464–5474. PMLR.

Köpf, A.; Kilcher, Y.; Von Rütte, D.; Anagnostidis, S.; Tam, Z. R.; Stevens, K.; Barhoum, A.; Nguyen, D.; Stanley, O.; Nagyfi, R.; et al. 2023. Openassistant conversations-democratizing large language model alignment. *Advances in neural information processing systems*, 36: 47669–47681.

Li, M.; Zhang, Y.; He, S.; Li, Z.; Zhao, H.; Wang, J.; Cheng, N.; and Zhou, T. 2024a. Superfiltering: Weak-to-Strong Data Filtering for Fast Instruction-Tuning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 14255–14273.

Li, M.; Zhang, Y.; Li, Z.; Chen, J.; Chen, L.; Cheng, N.; Wang, J.; Zhou, T.; and Xiao, J. 2023. From quantity to quality: Boosting llm performance with self-guided data selection for instruction tuning. *arXiv preprint arXiv:2308.12032*.

Li, Y.; Hui, B.; Xia, X.; Yang, J.; Yang, M.; Zhang, L.; Si, S.; Chen, L.-H.; Liu, J.; Liu, T.; et al. 2024b. One-Shot Learning as Instruction Data Prospector for Large Language Models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 4586–4601.

Liu, L.; Liu, X.; Wong, D. F.; Li, D.; Wang, Z.; Hu, B.; and Zhang, M. 2024. Selectit: Selective instruction tuning for large language models via uncertainty-aware self-reflection. *arXiv preprint arXiv:2402.16705*.

Liu, W.; Zeng, W.; He, K.; Jiang, Y.; and He, J. 2023a. What makes good data for alignment? a comprehensive study of automatic data selection in instruction tuning. *arXiv preprint arXiv:2312.15685*.

Liu, Y.; Tian, C. X.; Li, H.; Ma, L.; and Wang, S. 2023b. Neuron activation coverage: Rethinking out-of-distribution detection and generalization. *arXiv preprint arXiv:2306.02879*.

Longpre, S.; Hou, L.; Vu, T.; Webson, A.; Chung, H. W.; Tay, Y.; Zhou, D.; Le, Q. V.; Zoph, B.; Wei, J.; et al. 2023. The flan collection: Designing data and methods for effective instruction tuning. In *International Conference on Machine Learning*, 22631–22648. PMLR.

Ma, D.; Shang, G.; Chen, Z.; Qin, L.; Luo, Y.; Pan, L.; Fan, S.; Chen, L.; and Yu, K. 2025. Task-Specific Data Selection for Instruction Tuning via Monosemantic Neuronal Activations. *arXiv preprint arXiv:2503.15573*.

Mike, C.; Matt, H.; Ankit, M.; Jianwei, X.; Jun, W.; Sam, S.; Ali, G.; Patrick, W.; Matei, Z.; and Reynold, X. 2023. Free dolly: Introducing the world's first truly open instruction-tuned llm.

Mirzasoleiman, B.; Bilmes, J.; and Leskovec, J. 2020. Coresets for data-efficient training of machine learning models. In *International Conference on Machine Learning*, 6950–6960. PMLR.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35: 27730–27744.

Pruthi, G.; Liu, F.; Kale, S.; and Sundararajan, M. 2020. Estimating training data influence by tracing gradient descent. *Advances in Neural Information Processing Systems*, 33: 19920–19930.

Rai, D.; and Yao, Z. 2024. An Investigation of Neuron Activation as a Unified Lens to Explain Chain-of-Thought Eliciting Arithmetic Reasoning of LLMs. In *ACL (1)*.

Skean, O.; Arefin, M. R.; Zhao, D.; Patel, N.; Naghiyev, J.; LeCun, Y.; and Shwartz-Ziv, R. 2025. Layer by layer: Uncovering hidden representations in language models. *arXiv preprint arXiv:2502.02013*.

Srivastava, A.; Rastogi, A.; and et al, A. R. 2023. Beyond the Imitation Game: Quantifying and extrapolating the capabilities of language models. arXiv:2206.04615.

Team, Q. 2024. Qwen2.5: A Party of Foundation Models.

Toneva, M.; Sordoni, A.; Combes, R. T. d.; Trischler, A.; Bengio, Y.; and Gordon, G. J. 2018. An empirical study of example forgetting during deep neural network learning. *arXiv preprint arXiv:1812.05159*.

Wang, X.; Wen, K.; Zhang, Z.; Hou, L.; Liu, Z.; and Li, J. 2022. Finding Skill Neurons in Pre-trained Transformer-based Language Models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 11132–11152.

Wang, Y.; He, K.; Fu, D.; Gongque, Z.; Xu, H.; Chen, Y.; Wang, Z.; Fu, Y.; Dong, G.; Diao, M.; et al. 2024. How do your code llms perform? empowering code instruction tuning with really good data. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 14027–14043.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; Chi, E.; Le, Q. V.; Zhou, D.; et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35: 24824–24837.

Wei, Y.; Wang, Z.; Liu, J.; Ding, Y.; and Zhang, L. 2024. Magicoder: empowering code generation with OSS-INSTRUCT. In *Proceedings of the 41st International Conference on Machine Learning*, 52632–52657.

Xia, M.; Anastasopoulos, A.; Xu, R.; Yang, Y.; and Neubig, G. 2020. Predicting Performance for Natural Language Processing Tasks. In Jurafsky, D.; Chai, J.; Schluter, N.; and Tetreault, J., eds., *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 8625–8646. Online: Association for Computational Linguistics.

Xia, M.; Malladi, S.; Gururangan, S.; Arora, S.; and Chen, D. 2024. Less: Selecting influential data for targeted instruction tuning. *arXiv preprint arXiv:2402.04333*.

Xie, S. M.; Santurkar, S.; Ma, T.; and Liang, P. S. 2023. Data selection for language models via importance resampling. *Advances in Neural Information Processing Systems*, 36: 34201–34227.

Yu, Z.; Das, S.; and Xiong, C. 2024. Mates: Model-aware data selection for efficient pretraining with data influence models. *Advances in Neural Information Processing Systems*, 37: 108735–108759.

Zhang, Q.; Zhang, Y.; Wang, H.; and Zhao, J. 2024. Recost: External knowledge guided data-efficient instruction tuning. *arXiv preprint arXiv:2402.17355*.

Zhang, R.; Isola, P.; Efros, A. A.; Shechtman, E.; and Wang, O. 2018. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 586–595.

Zhao, Y.; Zhang, W.; Chen, G.; Kawaguchi, K.; and Bing, L. 2024. How do large language models handle multilingualism? *Advances in Neural Information Processing Systems*, 37: 15296–15319.

Zhao, Y.; Zhang, W.; Xie, Y.; Goyal, A.; Kawaguchi, K.; and Shieh, M. 2025. Understanding and enhancing safety mechanisms of LLMs via safety-specific neuron. In *The Thirteenth International Conference on Learning Representations*.

Zheng, Y.; Zhang, R.; Zhang, J.; Ye, Y.; Luo, Z.; Feng, Z.; and Ma, Y. 2024. 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.

Zhou, C.; Liu, P.; Xu, P.; Iyer, S.; Sun, J.; Mao, Y.; Ma, X.; Efrat, A.; Yu, P.; Yu, L.; et al. 2023. Lima: Less is more for alignment. *Advances in Neural Information Processing Systems*, 36: 55006–55021.

Zhou, H.; Tang, Y.; Qin, H.; Yang, Y.; Jin, R.; Xiong, D.; Han, K.; and Wang, Y. 2024. Star-agents: Automatic data optimization with llm agents for instruction tuning. *Advances in Neural Information Processing Systems*, 37: 4575–4597.

Zhu, S.; Pan, L.; Li, B.; and Xiong, D. 2024. LANDeRMT: Detecting and Routing Language-Aware Neurons for Selectively Finetuning LLMs to Machine Translation. *arXiv preprint arXiv:2409.19523*.