LAMDAS: LLM as an Implicit Classifier for Domain-specific Data Selection

Jian Wu[1,2*], **Hang Yu**[1*], **Bingchang Liu**[1], **Yang Wenjie**[1], **Peng Di**[1†], **Jianguo Li**[1], **Yue Zhang**[2†]

[1]Ant Group
[2]Westlake University
{wujian, zhangyue}@westlake.edu.cn, {hyu.hugo, bingchang.lbc, ywj439780, dipeng.dp, lijg.zero}@antgroup.com

Abstract

Adapting large language models (LLMs) to specific domains often faces a critical bottleneck: the scarcity of high-quality, human-curated data. While large volumes of unchecked data are readily available, indiscriminately using them for fine-tuning risks introducing noise and degrading performance. Strategic data selection is thus crucial, requiring a method that is both accurate and efficient. Existing approaches, categorized as similarity-based and direct optimization methods, struggle to simultaneously achieve these goals. In this paper, we introduce LAMDAS (*L*LM *A*s an i*M*plicit classifier for domain-specific *DA*ta *S*election), a novel approach that leverages the pre-trained LLM itself as an implicit classifier, thereby bypassing explicit feature engineering and computationally intensive optimization process. LAMDAS reframes data selection as a one-class classification problem, identifying candidate data that "belongs" to the target domain defined by a small reference dataset. Extensive experimental results demonstrate that LAMDAS not only exceeds the performance of full-data training using a fraction of the data but also outperforms nine state-of-the-art (SOTA) baselines under various scenarios. Furthermore, LAMDAS achieves the most compelling balance between performance gains and computational efficiency compared to all evaluated baselines.

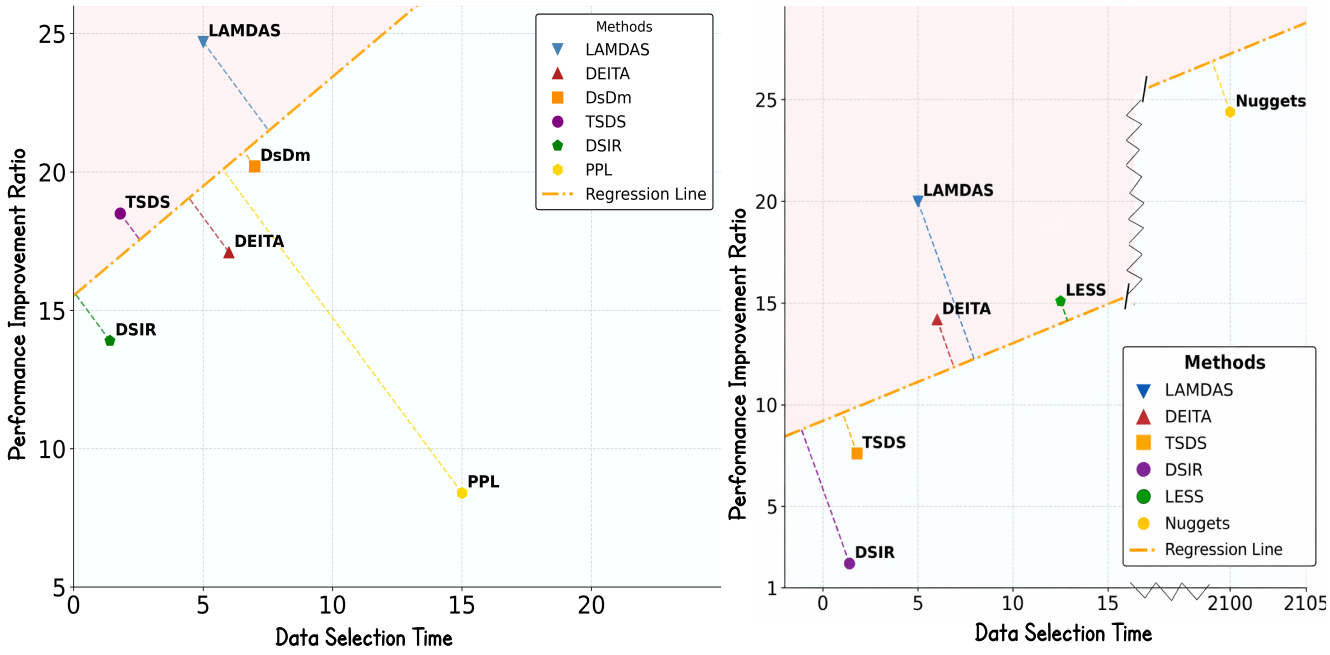
1 Introduction

The unprecedented capabilities of modern large language models (LLMs) rely on their ability to learn from vast and diverse datasets during pre-training, followed by adaptation to specialized domains or tasks via continual pre-training (CPT) and supervised fine-tuning (SFT). However, a key challenge arises when adapting LLMs to new scenarios: while high-quality, human-curated data is often scarce, large volumes of unchecked, automatically collected data, such as web-scraped content, crowd-sourced annotations, or synthetic examples, are readily accessible. Indiscriminately using this unchecked data for CPT or SFT carries the risk of introducing noise, misalignment with the target task, or even harmful patterns, leading to degraded model performance and increased hallucinations (Zhang et al. 2023; Li et al. 2023). Furthermore, training LLMs on unnecessarily large datasets incurs significant computational costs, requiring substantial GPU resources that are not readily available to all researchers and practitioners (He et al. 2024). It is therefore imperative to strategically select the most relevant training examples from the vast pool of unchecked candidate data, particularly when working with a small but meticulously curated reference dataset that accurately represents the target domain or task.

The core challenge in domain-specific data selection lies in identifying a method that is both **accurate** – maximizing LLM performance on the reference dataset after training on the selected data – and **efficient** – capable of processing massive candidate datasets, especially for CPT.

Unfortunately, existing approaches often struggle to achieve both of these goals simultaneously. As discussed in Section 2, the current literature on domain-specific data selection can be divided into two primary groups: similarity-based methods and direct optimization methods. Similarity-based methods typically extract features from both candidate and reference data, selecting candidate data based on their similarity to the reference data according to specific measures. Within this category, methods using easily obtained features like lexicon or embedding-based representations (Liu, Karbasi, and Rekatsinas 2024; Xie et al. 2023; Thrush, Potts, and Hashimoto 2025) and computationally cheap measures like cosine similarity or correlation are fast. However, selecting data based on such superficial characteristics can be ineffective or even detrimental to model performance, as pointed out in (Engstrom, Feldmann, and Madry 2024a). To tackle this issue, more recent methods utilize more informative features, such as gradients (Gu et al. 2025; Yu, Das, and Xiong 2024) or model weights (Xia et al. 2024; Huang et al. 2025), and employ advanced measures like optimal transport (Liu, Karbasi, and Rekatsinas 2024) and KL reduction (Xie et al. 2023), achieving improved performance gains at the cost of efficiency. On the other hand, direct optimization methods aim to enhance the LLM's performance on the reference data by selecting suitable subsets of candidate data based on frameworks like optimal control theory (Gu et al. 2025), data models (Engstrom, Feldmann, and Madry 2024a), conditional loss (Brandfonbrener et al. 2024), or Shapley values (He et al. 2024). However, the high computational burden of these methods typically requires them to estimate selection scores for only a representative subset of the candidate data.

[*]Equal contribution
[†]Corresponding author



(a) CPT (b) SFT

Figure 1: Performance gains versus data selection efficiency (selection time) for all methods in both (a) CPT and (b) SFT scenarios. A larger perpendicular distance, indicated by the dashed lines, from a method to the regression line—and above the regression line—indicates a more favorable trade-off. The regression line is estimated using linear regression on the x and y values of all methods—suggests a more favorable trade-off. The zigzag lines mean scale breaks.

A smaller or simplified model is then trained to extrapolate these scores to the entire candidate dataset, adding a layer of approximation that can degrade performance. In essence, these techniques often remain computationally expensive.

In this paper, we introduce **LAMDAS**, innovatively using **L**LM **A**s an i**M**plicit classifer for domain-specific **DA**ta **S**election. The proposed method is applicable to both data selection for CPT and SFT. Our approach effectively addresses the trade-off between performance and efficiency. The key insight is to reframe data selection as a one-class classification (OCC) problem: given a small reference dataset representing the target domain (positive samples), how can we identify candidate data that "belongs" to the same class or domain? Unlike prior works, LAMDAS leverages the pre-trained LLM itself as an **implicit** classifier, bypassing the need for explicit feature engineering or computationally intensive retraining. We first demonstrate that a prefix tuned on the reference dataset acts as a concise representation of the target domain – a "domain prefix". LAMDAS scores candidate examples by comparing their likelihood under the LLM **with** and **without** this domain prefix. Candidates exhibiting significantly higher likelihoods when conditioned on the prefix are prioritized, indicating a stronger alignment with the reference distribution. Since prefix tuning only modifies the prefix tokens while leaving the LLM's weights untouched, LAMDAS effectively preserves the LLM's general knowledge and prevents overfitting to the limited reference data. Moreover, given the relative simplicity of binary classification, a smaller LLM can often achieve good classification performance (as shown in our experiments), thereby ensuring the efficiency of our approach.

Extensive experimental results demonstrate that LAMDAS not only surpasses full data training while utilizing only a small fraction of the data but also outperforms **seven** SOTA baselines for CPT data selection and **nine** baselines for SFT data selection. Moreover, as depicted in Figure 1, LAMDAS achieves **the most favorable trade-off** between performance gains and efficiency among all baselines evaluated. In summary, the key contributions of this work are[1]:

- We innovatively formulate the domain-specific data selection problem as an OCC problem.
- We propose leveraging small LLMs as implicit classifiers to address the OCC problem, striking a compelling balance between efficacy and efficiency.
- We demonstrate through extensive experimentation on both CPT and SFT data selection tasks that LAMDAS outperforms other SOTA baselines.

2 Related Works

As mentioned in the introduction, domain-specific data selection can be broadly categorized into two groups: similarity-based methods and direct optimization methods.

[1]Code is available from https://github.com/codefuse-ai/Lamdas

2.1 Similarity-based Methods

Similarity-based methods typically involve three primary steps: extracting features from both candidate and reference datasets, measuring the similarity between these two sets, and selecting candidate samples that closely match the reference data. Early approaches often relied on simple features and measures, including lexicon-based features (Silva and Barbosa 2024), embedding representations (Liu, Karbasi, and Rekatsinas 2024), and human-crafted rules (Wettig et al. 2024; Sachdeva et al. 2024). The similarity measures used in these methods include binary grammar discriminators (Sachdeva et al. 2024; Touvron et al. 2023), rules defined by LLMs (Li et al. 2024), cosine similarity (Rubin, Herzig, and Berant 2022), and perplexity (Marion et al. 2023; Muennighoff et al. 2023). Although these methods are efficient and impose low computational overhead, they tend to capture only superficial features of candidate data and fail to account for the intricate relationships between candidate and reference datasets. As pointed out in (Engstrom, Feldmann, and Madry 2024a), reliance on simplistic features and measures can adversely affect model performance. To address these limitations, researchers have investigated ways to enhance the complexity of either features or similarity measures. For example, some methods incorporate gradients (Evans et al. 2024; Everaert and Potts 2024; Zhao et al. 2025), as the gradient of selected data is expected to align with that of the reference data, thereby ensuring that loss on the reference data decreases upon training with the selected samples. An alternative approach, Grad-Mimic (Huang et al. 2025), selects data that aligns its gradient with a direction that points toward the reference model in weight space. However, computing gradients can be computationally expensive, which poses scalability challenges for these methods. To alleviate this issue, SkMM (Dong et al. 2024) employs gradient sketching, while LESS (Xia et al. 2024) uses low-rank adaptation (LoRA) gradients as a low-dimensional approximation of the original gradient. MATES (Yu, Das, and Xiong 2024) instead trains a small model to approximate the mapping between the candidate data and its influence score. Apart from using more advanced features, more effective measures have also been explored, such as optimal transport (Kaplan et al. 2020; Liu, Karbasi, and Rekatsinas 2024) and KL reduction (Xie et al. 2023). These techniques seek to align the distribution between the selected and the reference in the feature space, improving performance at the expense of efficiency.

2.2 Direct Optimization Methods

Direct optimization methods allow for direct interaction between the candidate and reference data, with the goal of maximizing the performance on the reference dataset when a model is trained using the selected data. SHED (He et al. 2024) uses the Shapley value framework to attribute loss reduction to the candidate data, requiring training the model on different subsets of the candidate data and then computing the attribution score. Due to the extreme computational cost, SHED must first cluster the candidate data and then compute the attribution score for each cluster based on the score of the data point at the cluster center. Alternatively, PDS (Gu et al. 2025) formulates data selection as an optimal control problem, using Pontryagin's Maximum Principle (PMP) to derive necessary conditions for optimal data selection. However, solving the PMP equations is also computationally expensive, so PDS only solves the equation for representative samples within the candidate set and then approximates the solution using a small model. The resulting small model is then used for data selection, which may introduce approximation errors. Similarly, DsDm (Engstrom, Feldmann, and Madry 2024a) frames data selection as an optimization problem using "data models" to approximate how the learning algorithm uses training data to make predictions on the target tasks. In addition, Nuggets (Li et al. 2023) selects candidates that increase the generation probability of the reference data by comparing the likelihood of the reference data with and without the candidate as a one-shot example in the prompt, requiring a comparison of every candidate-reference data pair. In summary, these methods can improve performance by enabling direct interaction between candidate and reference data, but they often suffer from high computational cost.

In contrast to the above two categories, our proposed LAMDAS method aims to increase efficiency while still allowing for meaningful interaction between the candidate and reference data to ensure strong performance. We will elaborate on the relationship between LAMDAS and existing methods in Section 3.2 after we formally introduce our methodology.

3 Methodology

Let $\mathcal{D}_{ref} = \{\boldsymbol{x}_i\}_{i=1}^N$ denote the reference dataset and $\mathcal{D}_{cand} = \{\boldsymbol{y}_j\}_{j=1}^M$ denote the candidate dataset. We frame the data selection problem as a classification task, where our objective is to train a classifier to determine whether each sample in the candidate dataset should be retained. However, a significant challenge arises in that we possess only positive samples from the reference dataset during training. This scenario defines the problem as a one-class classification (OCC) problem, where we must identify relevant samples without negative examples for guidance.

3.1 LLM as an Implicit Classifier

In the natural language processing domain, classification tasks typically utilize one of two frameworks: BERT-based encoders or GPT-based decoders. BERT-based methods typically fine-tune the model for classification, but they require negative samples, making them unsuitable for the OCC setting. Conversely, GPT-based methods, as exemplified by Ask-LLM (Sachdeva et al. 2024), could potentially select data by prompting the text decoder with a natural language description of the class and asking whether the specific sample belongs to this class. However, in our scenario, we lack a clear natural language description of the target domain; we only have the representative examples in the reference data. Summarizing the reference data into a concise and accurate textual description can be difficult because text summarization inevitably loses some information through the bottleneck of human-interpretable natural language. Furthermore, some domains, such as emerging slang dialects, exhibit patterns that are difficult to capture in formal language but manifest

clearly in data. To circumvent the above limitations, we propose to harness the LLM itself as an implicit classifier.

Prefix Tuning for Domain Representation Learning We represent the target domain $\boldsymbol{C}$ (defined by $\mathcal{D}_{ref}$) via a learnable domain prefix $\boldsymbol{C}$, which steers the LLM's generation toward the reference distribution. Specifically, we learn a soft prefix $\boldsymbol{C}$ that maximizes the likelihood of the reference data:

$$\begin{aligned}\boldsymbol{C} &= \arg\max \log p(\boldsymbol{x}_{1:N}|\boldsymbol{C}) \\ &= \arg\max \sum_{i=1}^{N} \log p(\boldsymbol{x}_i|\boldsymbol{C}). \end{aligned} \tag{1}$$

Here we opt for prefix tuning due to its greater flexibility compared to prompt tuning. By employing a learned soft prefix, we avoid the need to manually craft domain descriptions, a requirement that hinders the use of Ask-LLM. Furthermore, the above estimation inherently maximizes the mutual information between the reference data and the learned domain representation $\boldsymbol{C}$. Given the typically small size of the reference dataset, the prefix can effectively retain essential information about the domain defined by that dataset.

Likelihood Ratio for Data Selection Once we have learned the domain representation $\boldsymbol{C}$, we can estimate the probability that a candidate sample belongs to the same class or domain, leveraging Bayes' theorem:

$$p(\boldsymbol{C}|\boldsymbol{y}_j) = \frac{p(\boldsymbol{y}_j|\boldsymbol{C})p(\boldsymbol{C})}{p(\boldsymbol{y}_j)}. \tag{2}$$

Here, we can disregard the prior $p(\boldsymbol{C})$ because it remains constant across all samples $\boldsymbol{y}_j$ in the candidate set, and does not assist in differentiating between selected and unselected data. Consequently, we define the sample selection score using the likelihood ratio of $\boldsymbol{y}_j$ from the LLM with and without the prefix or class representation $\boldsymbol{C}$:

$$s(\boldsymbol{y}_j) = \frac{p(\boldsymbol{y}_j|\boldsymbol{C})}{p(\boldsymbol{y}_j)}. \tag{3}$$

We select $\boldsymbol{y}_j$ when the score exceeds the threshold $\tau = 1$, thus prioritizing samples that align more closely with the representation of the reference distribution. Indeed,

Proposition 1. *(Likelihood Ratio as Domain Discriminant): The score $s(\boldsymbol{y}_j)$ is equivalent to the likelihood ratio test statistic for distinguishing $p(\boldsymbol{y}_j|\boldsymbol{C})$ from $p(\boldsymbol{y}_j)$. This likelihood ratio test is optimal for binary hypothesis testing with fixed Type I or Type II error rates.*

Proof. This proposition is supported by the Neyman-Pearson lemma (Casella and Berger 2024). □

The formulation presented leverages the LLM as an implicit classifier without modifying the LLM's weights. This strategy preserves the rich pre-trained knowledge embedded in the LLM, ensuring that our method is generalizable to unseen samples within the candidate set. Additionally, our approach allows for seamless application across different domains by simply altering the domain prefix. Given that classification tasks are relatively easy, even small LLMs can perform well in practice, further enhancing efficiency.

3.2 Relation to Other Methods

Balancing Efficacy and Efficiency As previously discussed, strategies for domain-specific data selection can be categorized into two main groups: similarity-based and direct optimization methods. In fact, similarity-based methods can be viewed as bi-encoder or dual-encoder approaches in information retrieval (Muennighoff 2022), where features for both candidate and reference data are independently extracted via encoders and then compared using similarity measures. This constrained interaction limits their ability to fully capture relationships between the two data sets, often resulting in suboptimal performance. In contrast, direct optimization methods facilitate direct interaction between candidate and reference data, akin to cross-encoder approaches (Liao et al. 2024), thus improving performance. For example, NUGGETS allows for interaction between every pair $(\boldsymbol{x}_i, \boldsymbol{y}_j)$ by computing the likelihood ratio of $p(\boldsymbol{y}_j|\boldsymbol{x}_i)/p(\boldsymbol{x}_i)$. However, such methods often significantly increase computational costs.

Our proposed method seeks to strike a balance between these two approaches. Initially, we condense the reference data into a concise domain representation or prefix, which then allows for direct interaction between all candidate data and this domain representation. The maximum likelihood estimation of the domain representation (see Eq. (1)) theoretically maximizes the mutual information between the domain prefix and the reference data. Given that reference datasets typically have limited sample sizes, it is straightforward for the prefix to retain critical information. The direct interaction between the prefix and each candidate sample ensures that performance exceeds that of similarity-based methods, as cross-encoder methods typically deliver superior outcomes compared to bi-encoder techniques. Moreover, the concise domain prefix eliminates the need for exhaustive interactions between every candidate and reference data pair, thus enhancing efficiency in comparison to direct optimization methods.

Connection to Classifier-Free Guidance Our method draws inspiration from classifier-free guidance employed in diffusion models (Yang et al. 2022), which uses a generative diffusion model as an implicit classifier to steer the text-to-image generation process, ensuring that generated images can be classified into the domains defined by provided text. In our case, we utilize the LLM as an implicit classifier for data selection, ensuring that the selected subset from the candidate dataset accurately reflects the domain defined by the reference data.

4 Experiments

We conduct comprehensive experiments to validate LAMDAS across continued pre-training (CPT) and supervised fine-tuning (SFT) scenarios. Our experiments aim to answer the following questions:

1. **Performance Gains**: Does LAMDAS outperform existing methods in downstream tasks when selecting the same amount of data?
2. **Efficiency**: Does LAMDAS achieve lower time complexity for data selection compared with existing methods?

3. **Sensitivity**: Is LAMDAS's performance sensitive to the selection threshold τ, the length of the prefix, the size of the classifier, and the base LLM for the classifier?

To answer our research questions, we design a comprehensive experimental setup for both coding and mathematical reasoning across CPT and SFT scenarios. We use high-quality reference data to select from large candidate pools and evaluate performance on a suite of relevant benchmarks chosen to test generalization. LAMDAS is benchmarked against several groups of baselines: seven methods applicable to both CPT and SFT, including similarity-based methods (DSIR, TSDS), direct optimization methods (DsDm), and perplexity-based (PPL) approaches; two additional direct optimization methods for the smaller SFT datasets (Nuggets, LESS) that are too computationally intensive for CPT; and standard controls (Random, Full). For brevity, detailed specifications of the datasets, baseline implementations, and hyperparameters are provided in Appendix.

The next two subsections present the evaluation results comparing LAMDAS with several baseline methods. Our experiments focus on assessing the capabilities of the Qwen2.5 series models when trained using datasets selected by LAMDAS and the baseline methods. We examine performance on coding tasks for CPT, and on both coding and mathematical reasoning tasks for SFT.

4.1 Data Selection for CPT

The results demonstrate that CPT with data selected by LAMDAS consistently achieves the best performance across all metrics, regardless of the underlying model. Indeed, when using the selected data to train the Qwen2.5-7B model, LAMDAS outperforms the same model trained on the "Full" data by an average of 24.7%, while simultaneously reducing the dataset size by a significant 75%. Similarly, Qwen2.5-0.5B and Qwen-1.5B resulting from LAMDAS also outperform the same models trained on the "Full" data by an average of 52.6% and 46.1%. As visualized in Figure 2, LAMDAS effectively concentrates on the region that closely matches the reference distribution, which explains its superior performance in downstream tasks. The second-best method, DsDm, focuses on direct optimization to minimize loss on the reference set by selecting candidate data based on influence scores. However, its high computational cost necessitates the use of a simplified proxy estimator to approximate these scores, trading approximation error for efficiency (Engstrom, Feldmann, and Madry 2024b).

The third-best method, TSDS, employs text embeddings as features and uses optimal transport as a measure of similarity. TSDS's better performance compared to DSIR illustrates the importance of careful feature selection in similarity-based approaches, as DSIR relies solely on superficial features like n-grams. Despite this, both similarity-based methods still fall short of LAMDAS's performance. This can be attributed to the fact that feature extraction and similarity computation, even with advanced features and measures, do not offer the same flexibility as LAMDAS. LAMDAS facilitates direct interaction between candidate and reference domains through a cross-attention mechanism in the LLM, creating connections between the domain prefix and candidate samples.

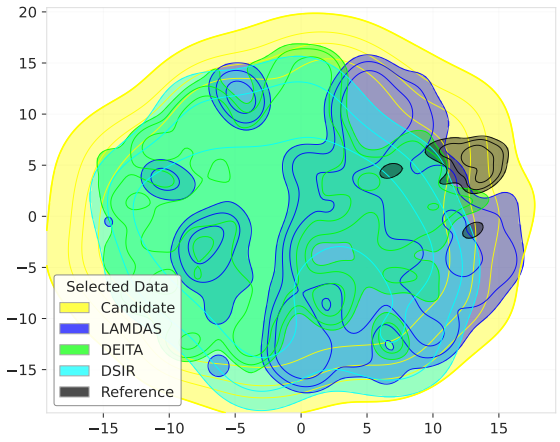


Figure 2: Distribution comparison of candidate data (yellow), LAMDAS-selected data (blue), DEITA-selected data (green), DSIR-selected data (cyan), and reference data (black). Density contours represent the 10-50% highest density regions for each distribution. Embeddings are normalized, and reduced to 2D space via t-SNE.

In contrast, DEITA selects data based on quality and complexity scores, prioritizing factors other than domain relevance. This explains its underperformance on domain-specific tasks. While PPL may implicitly consider domain relevance, it primarily selects data where next tokens are easily predictable given previous tokens, driven by its objective of minimizing next-token prediction loss (see Table 6 in the appendix). Interestingly, the "Random" baseline outperforms the "Full" baseline in our CPT experiments. We attribute this to the nature of the CPT data, which consists of pure code data from GitHub and The Stack v1. Continuously pre-training with this entire dataset can lead to catastrophic forgetting of the model's NLP capabilities (as shown in Table 1). The evaluation benchmarks, however, require instruction following for code completion and text-to-code tasks. The "Random" baseline mitigated this issue by only pre-training the model with 15B tokens, thus partially preserving the model's NLP abilities and leading to better results. Furthermore, other data selection methods further improved the results of "Random", showcasing their effectiveness.

4.2 Data Selection for SFT

Code Once again, as shown in table 2, the Qwen2.5-32B-Instruct model trained on data selected by LAMDAS demonstrates superior performance, exceeding the "Full" method by 20.0% and outpacing the second-best method, LESS, by an average of 15.1%. LESS and DsDm perform comparably, achieving the second and third best results, as both methods aim to select training data that minimizes the model's loss on the reference data. However, both face challenges related to high time complexity due to the expensive computation of gradients, as discussed in Section 4.3. Additionally, LESS suffers from the limitations inherent in similarity-based methods, while DsDm improves efficiency by using small proxy models for gradient computation, though this can introduce

Methods	Data Size	Models	HE	HE+	MBPP	MBPP+	LCB	CRUXEval	AVG
LAMDAS(ours)	15B	Qwen2.5-0.5B	**23.5**	**19.4**	**43.9**	**37.8**	**5.8**	**4.9**	**23.5**+52.6%
		Qwen2.5-1.5B	**26.7**	**22.3**	**50.5**	**44.3**	**13.7**	**9.7**	**27.9**+46.1%
		Qwen2.5-7B	**35.2**	**29.2**	**58.5**	**50.6**	**25.7**	**15.8**	**35.8**+24.7%
DEITA (Liu et al. 2024a)	15B	Qwen2.5-0.5B	22.1	17.6	42.5	35.9	4.1	3.9	21.5+39.6%
		Qwen2.5-1.5B	25.8	21.2	48.7	43.1	9.9	8.3	26.4+38.2%
		Qwen2.5-7B	34.1	27.8	56.1	49.2	21.3	14.9	33.6+17.1%
DsDm(Engstrom, Feldmann, and Madry 2024a)	15B	Qwen2.5-0.5B	23.2	18.6	39.3	35.4	4.9	4.2	20.9+35.7%
		Qwen2.5-1.5B	25.8	21.5	47.6	43.8	12.5	8.7	26.7+39.8%
		Qwen2.5-7B	34.7	28.2	55.8	49.4	23.5	15.2	34.5+20.2%
TSDS (Liu, Karbasi, and Rekatsinas 2024)	15B	Qwen2.5-0.5B	22.9	19.1	41.3	36.4	5.3	4.6	21.6+40.3%
		Qwen2.5-1.5B	26.2	21.9	49.8	44.1	13.4	8.8	27.4+43.5%
		Qwen2.5-7B	34.2	27.2	54.7	48.6	24.2	15.3	34.0+18.5%
DSIR (Xie et al. 2023)	15B	Qwen2.5-0.5B	21.5	17.3	41.2	34.8	3.89	3.5	20.4+32.5%
		Qwen2.5-1.5B	24.2	20.1	46.3	41.8	9.1	7.1	24.8+29.8%
		Qwen2.5-7B	33.5	26.3	53.1	47.8	20.9	12.5	32.7+13.9%
PPL	15B	Qwen2.5-0.5B	20.7	15.4	39.2	33.7	3.6	3.4	19.3+25.3%
		Qwen2.5-1.5B	22.1	19.4	45.7	41.4	10.2	6.6	24.2+26.7%
		Qwen2.5-7B	30.6	27.5	51.7	46.3	19.2	11.3	31.1+8.4%
Random	15B	Qwen2.5-0.5B	19.7	14.8	38.6	32.4	3.41	3.3	18.7+21.4%
		Qwen2.5-1.5B	21.7	17.9	43.7	38.9	8.6	6.5	22.9+19.9%
		Qwen2.5-7B	29.5	26.1	49.2	44.7	18.3	9.7	29.6+3.1%
Full	100B	Qwen2.5-0.5B	12.8	9.3	35.1	28.6	3.3	3.0	15.4
		Qwen2.5-1.5B	17.8	13.2	38.1	30.3	7.9	7.2	19.1
		Qwen2.5-7B	27.2	22.1	52.9	43.4	16.4	9.9	28.7
Base	0B	Qwen2.5-0.5B	16.5	12.3	37.1	30.6	5.2	6.0	18.0
		Qwen2.5-1.5B	21.2	17.3	41.8	37.9	8.1	7.7	21.1
		Qwen2.5-7B	28.9	22.1	45.9	40.4	14.4	8.9	26.8

Table 1: Pass@1 performance on HumanEval (HE), MBPP, LCB, and CRUXEval. All models are continually pretrained on selected data and evaluated with a zero-shot prompt strategy. We also compute the average of models' performance on all benchmarks and calculate the proportion of improvement compared with the "Full" data. The best results are highlighted in **bold**, the second-best results are underlined, and the third are wavy underlined.

approximation error. Consequently, both methods perform worse than LAMDAS. Similarly, although Nuggets is a direct optimization method, it still underperforms relative to LAMDAS. This shortfall may stem from Nuggets' reliance on the original Llama2-7B model to assess the relevance of candidate samples to the reference data, which may not fully capture the nuances of the specific domain. Conversely, LAMDAS utilizes prefix tuning for domain adaptation before evaluating candidate samples for relevance, enhancing its effectiveness. Different from the CPT experiment, "Full" performs better than "Random" here. This may be because all candidate data are related to coding tasks with natural language instructions, eliminating the risk of catastrophic forgetting of instruction-following capabilities when training with purely code data.

Math The performance trends for the math reasoning task are similar to those in coding tasks; detailed discussion is provided in Appendix.

4.3 Efficiency

As shown in Table 3 in Appendix, LAMDAS is only slower than the similarity-based methods TSDS and DSIR that are based on simple features such as text embeddings or n-grams, but is faster than all the remaining baselines. Indeed, with a time complexity of 2× model forward passes ($C_{forward}$) and an execution speed of 5 seconds per 100 samples, LAMDAS with Qwen2.5-Coder-0.5B model as the reference model achieves a 1.4× speedup over direct optimization method like DsDm (0.5B model, 7s/100) and 2.5× over the similarity-based method LESS (0.5B model, 12.5s/100). The table also illustrates that the actual computational time aligns with the theoretical time complexity and model size for data selection.

Furthermore, to directly compare methods based on their performance gains and efficiency, we draw inspiration from Liu (Liu et al. 2024b), positing that methods with higher computational complexity should yield better data selection and, consequently, greater performance gains. To illustrate this trend, we apply linear regression to derive a regression line that represents the relationship between performance gains and runtime across all methods, as shown in Figure 1. We then compute the perpendicular distances (represented by the dashed lines in Figure 1) from each method to the regression line. A larger perpendicular distance above the regression line indicates a better trade-off between performance gains and efficiency.

Figure 1 clearly illustrates that LAMDAS achieves a more favorable trade-off between performance gains and efficiency in both the CPT and SFT scenarios. While the two fastest methods, TSDS and DSIR, exhibit lower computational over-

Methods	Data Size	Models	HE	HE+	MBPP	MBPP+	LCB	CRUXEval	AVG
LAMDAS(ours)	750K	Qwen2.5-0.5B-Instruct	**25.3**	**23.1**	**48.2**	**42.1**	**8.7**	**15.1**	**27.1**+36.9%
		Qwen2.5-1.5B-Instruct	**32.8**	**30.1**	**55.6**	**47.1**	**17.2**	**25.5**	**34.3**+30.3%
		Qwen2.5-7B-Instruct	**42.8**	**37.6**	**62.1**	**55.3**	**32.5**	**37.2**	**44.6**+19.6%
		Qwen2.5-32B-Instruct	**89.0**	**84.1**	**75.3**	**69.4**	**44.1**	**52.3**	**69.0**+20.0%
		Llama3-8B-Instruct	**71.9**	**66.7**	**64.3**	**59.1**	**29.8**	**49.5**	**56.9**+13.6%
DSIR(Xie et al. 2023)	750K	Qwen2.5-0.5B-Instruct	21.2	18.5	36.7	31.5	5.5	11.2	20.8+5.1%
		Qwen2.5-1.5B-Instruct	27.8	23.1	43.2	35.4	13.6	17.6	26.8+1.5%
		Qwen2.5-7B-Instruct	38.9	32.1	52.3	47.6	24.4	30.3	37.6+0.8%
		Qwen2.5-32B-Instruct	80.3	72.7	61.9	56.3	40.5	47.3	59.8+2.2%
		Llama3-8B-Instruct	65.1	61.4	60.8	55.6	23.2	46.8	52.2+4.2%
TSDS (Liu, Karbasi, and Rekatsinas 2024)	750K	Qwen2.5-0.5B-Instruct	23.4	21.5	40.7	34.5	6.8	11.9	23.1+16.7%
		Qwen2.5-1.5B-Instruct	30.8	25.2	49.3	43.2	14.8	18.3	30.3+14.8%
		Qwen2.5-7B-Instruct	40.1	34.2	55.6	50.2	27.3	32.1	39.9+7.0%
		Qwen2.5-32B-Instruct	82.5	75.9	63.4	59.2	41.2	48.9	61.9+7.6%
		Llama3-8B-Instruct	67.2	62.5	61.3	57.0	24.5	45.7	53.0+5.8%
DEITA (Liu et al. 2024a)	750K	Qwen2.5-0.5B-Instruct	23.7	21.8	46.1	40.5	7.3	13.5	25.5+28.8%
		Qwen2.5-1.5B-Instruct	29.3	25.2	51.4	42.9	15.7	21.1	30.9+17.0%
		Qwen2.5-7B-Instruct	39.8	33.2	59.7	51.8	29.3	34.7	41.4+11.0%
		Qwen2.5-32B-Instruct	86.1	80.2	72.9	65.9	41.7	48.1	65.7+14.2%
		Llama3-8B-Instruct	67.8	63.1	62.1	57.2	24.1	45.2	53.3+6.4%
DsDm(Engstrom, Feldmann, and Madry 2024a)	750K	Qwen2.5-0.5B-Instruct	24.2	21.7	47.5	41.8	7.6	14.3	26.2 +32.3%
		Qwen2.5-1.5B-Instruct	31.4	29.5	51.2	42.3	15.5	24.1	32.3+22.3%
		Qwen2.5-7B-Instruct	39.8	34.7	60.3	50.8	29.6	36.2	41.9 +12.3%
		Qwen2.5-32B-Instruct	87.2	81.1	73.5	67.1	42.9	47.6	66.6+15.8%
		Llama3-8B-Instruct	69.3	61.6	63.2	58.1	25.7	46.4	54.1+8.0%
Nuggets(Li et al. 2023)	750K	Qwen2.5-0.5B-Instruct	24.1	22.7	43.4	39.4	7.7	14.8	25.4+28.3%
		Qwen2.5-1.5B-Instruct	29.8	24.9	51.7	42.9	15.4	23.9	31.4+18.9%
		Qwen2.5-7B-Instruct	38.7	33.6	57.5	52.1	29.8	37.0	41.5+11.3%
		Qwen2.5-32B-Instruct	86.4	80.0	72.9	65.6	42.1	47.5	65.8+14.4%
		Llama3-8B-Instruct	70.2	64.7	61.9	56.8	26.7	47.4	54.6+9.0%
LESS (Xia et al. 2024)	750K	Qwen2.5-0.5B-Instruct	24.5	22.3	48.2	43.1	8.4	13.8	26.7+34.8%
		Qwen2.5-1.5B-Instruct	30.2	26.1	53.2	44.8	15.8	22.8	32.2 +3.1%
		Qwen2.5-7B-Instruct	40.2	35.1	61.1	54.2	30.9	35.7	42.9+15.0%
		Qwen2.5-32B-Instruct	86.7	80.5	73.1	66.2	42.3	48.4	66.2 +15.1%
		Llama3-8B-Instruct	71.2	65.3	63.1	58.2	27.3	48.1	55.5+10.8%
Random	750K	Qwen2.5-0.5B-Instruct	18.6	15.8	32.5	28.7	5.4	8.1	18.2-0.8%
		Qwen2.5-1.5B-Instruct	24.5	21.2	37.8	33.1	14.0	15.3	24.3-0.8%
		Qwen2.5-7B-Instruct	35.6	30.3	49.2	43.2	23.5	27.9	35.0-0.6%
		Qwen2.5-32B-Instruct	76.2	68.9	56.7	51.2	35.1	39.8	54.7-4.9%
		Llama3-8B-Instruct	62.1	59.2	55.4	51.5	17.8	40.3	47.7 -4.8%
Full	1800K	Qwen2.5-0.5B-Instruct	20.1	16.5	34.1	31.6	4.4	12.3	19.8
		Qwen2.5-1.5B-Instruct	27.4	22.5	40.2	35.2	12.7	20.6	26.4
		Qwen2.5-7B-Instruct	37.6	31.8	51.2	45.6	25.2	32.2	37.3
		Qwen2.5-32B-Instruct	77.5	70.4	60.1	54.3	38.5	44.2	57.5
		Llama3-8B-Instruct	65.7	60.1	57.9	53.2	20.1	43.5	50.1
Base	0K	Qwen2.5-0.5B-Instruct	15.2	10.5	26.7	21.9	3.6	6.7	14.1
		Qwen2.5-1.5B-Instruct	22.3	17.6	35.1	31.4	8.9	15.4	21.8
		Qwen2.5-7B-Instruct	31.1	25.8	42.1	35.7	19.5	24.8	29.8
		Qwen2.5-32B-Instruct	72.1	65.5	53.4	47.9	30.4	37.1	51.1
		Llama3-8B-Instruct	62.8	57.3	54.8	50.5	14.2	36.2	46.0-8.2%

Table 2: Performance of Qwen2.5 serious models trained on various data selection methods on HumanEval (HE), MBPP, LiveCodeBench, and CRUXEval (Zero-shot, Pass@1). We also compute the average of models' performance on all benchmarks and calculate the proportion of improvement compared with the "Full" data. The best results are highlighted in **bold**, the second-best results are underlined, and the third are highlighted.

head, the data they select does not improve model performance to the same extent as the data selected by LAMDAS. Although the remaining methods may achieve better performance than TSDS and DSIR, they do so at the expense of efficiency. For example, LAMDAS can filter 100 billion tokens in 35 hours using only eight A100 GPUs, whereas LESS requires more than 144 hours for the same task. Moreover, the data selected by LESS fails to improve model performance as effectively as the data selected by LAMDAS. Thus, LAMDAS surpasses these methods in terms of both performance gains and efficiency.

4.4 Ablation Studies

We also conduct a series of ablation studies to validate the impact of selection threshold τ, prefix length, data selection model size, and data selection model type on the performance of LAMDAS. Due to the page limit, we summarize the major findings here and provide a detailed analysis in Appendix . The experimental results reveal four key insights: 1) the optimal value of $\tau = 1$ aligns with our design principle: to select candidate samples whose likelihood increases when conditioned on the domain prefix, thus ensuring relevance; 2) The prefix length of 30 represents an optimal "elbow point",

balancing representation and efficiency; 3) the binary classification for data selection is a relatively simple task that can be effectively handled by small LLMs; 4) The domain specific models typically result in a more effective classifier.

5 Conclusion

In this paper, we propose LAMDAS, a novel domain-specific data selection method that utilizes a pre-trained LLM itself as an implicit classifier. By reframing data selection as a one-class classification (OCC) problem, LAMDAS circumvents the need for explicit feature engineering, common in similarity-based methods, and the computationally intensive retraining required by direct optimization methods. Experimental results demonstrate that LAMDAS achieves a superior balance between effectiveness and efficiency compared to nine SOTA baselines. Notably, models continually pretrained on data selected by LAMDAS, using a 15% selection ratio, outperform those trained on the full dataset by an average of at least 20%. It is important to note that an ideal dataset typically exhibits high quality, broad coverage of the target domain, and high diversity, and may also introduce novel knowledge to the model, rather than simply containing samples already encountered during initial training. While various data selection methods have been proposed to address each of these aspects individually, LAMDAS can be readily integrated with these methods in a pipeline architecture to further enhance the domain-specific nature of the selected dataset. In future work, we aim to develop such pipelines for comprehensive data selection.

Acknowledgement

This work was supported by Ant Group Research Intern Program.

References

Brandfonbrener, D.; Zhang, H.; Kirsch, A.; Schwarz, J. R.; and Kakade, S. 2024. Color-filter: Conditional loss reduction filtering for targeted language model pre-training. *Advances in Neural Information Processing Systems*, 37: 97618–97649.

Casella, G.; and Berger, R. 2024. *Statistical inference*. CRC press.

Dong, Y.; Phan, V. H.; Pan, X.; and Lei, Q. 2024. Sketchy moment matching: Toward fast and provable data selection for finetuning. *Advances in Neural Information Processing Systems*, 37: 43367–43402.

Engstrom, L.; Feldmann, A.; and Madry, A. 2024a. DsDm: Model-Aware Dataset Selection with Datamodels. In *International Conference on Machine Learning*, 12491–12526. PMLR.

Engstrom, L.; Feldmann, A.; and Madry, A. 2024b. DsDm: Model-Aware Dataset Selection with Datamodels. In *International Conference on Machine Learning*, 12491–12526. PMLR.

Evans, N. J.; Mills, G. B.; Wu, G.; Song, X.; and McWeeney, S. K. 2024. Data Valuation with Gradient Similarity. *ArXiv*.

Everaert, D.; and Potts, C. 2024. GIO: Gradient Information Optimization for Training Dataset Selection. In *The Twelfth International Conference on Learning Representations*.

Gu, Y.; Dong, L.; Wang, H.; Hao, Y.; Dong, Q.; Wei, F.; and Huang, M. 2025. Data Selection via Optimal Control for Language Models. In *The Thirteenth International Conference on Learning Representations*.

He, Y.; Wang, Z.; Shen, Z.; Sun, G.; Dai, Y.; Wu, Y.; Wang, H.; and Li, A. 2024. SHED: Shapley-Based Automated Dataset Refinement for Instruction Fine-Tuning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Huang, T.-H.; Bilkhu, M.; Sala, F.; and Movellan, J. 2025. Evaluating Sample Utility for Data Selection by Mimicking Model Weights. *ArXiv*, abs/2501.06708.

Kaplan, J.; McCandlish, S.; Henighan, T.; Brown, T. B.; Chess, B.; Child, R.; Gray, S.; Radford, A.; Wu, J.; and Amodei, D. 2020. Scaling Laws for Neural Language Models. *ArXiv*, abs/2001.08361.

Li, X.; Gao, M.; Zhang, Z.; Yue, C.; and Hu, H. 2024. Rule-based Data Selection for Large Language Models. *ArXiv*, abs/2410.04715.

Li, Y.; Hui, B.; Xia, X.; Yang, J.; Yang, M.; Zhang, L.; Si, S.; Chen, L.-H.; Liu, J.; Liu, T.; Huang, F.; and Li, Y. 2023. One Shot Learning as Instruction Data Prospector for Large Language Models. In *Annual Meeting of the Association for Computational Linguistics*.

Liao, Z.; Yu, H.; Li, J.; Wang, J.; and Zhang, W. 2024. D2LLM: Decomposed and Distilled Large Language Models for Semantic Search. 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)*, 14798–14814. Bangkok, Thailand: Association for Computational Linguistics.

Liu, W.; Zeng, W.; He, K.; Jiang, Y.; and He, J. 2024a. What Makes Good Data for Alignment? A Comprehensive Study of Automatic Data Selection in Instruction Tuning. In *The Twelfth International Conference on Learning Representations*.

Liu, Z.; Karbasi, A.; and Rekatsinas, T. 2024. TSDS: Data Selection for Task-Specific Model Finetuning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Liu, Z.; Ke, R.; Liu, Y.; Jiang, F.; and Li, H. 2024b. Take the essence and discard the dross: A rethinking on data selection for fine-tuning large language models. *arXiv preprint arXiv:2406.14115*.

Marion, M.; Üstün, A.; Pozzobon, L.; Wang, A.; Fadaee, M.; and Hooker, S. 2023. When Less is More: Investigating Data Pruning for Pretraining LLMs at Scale. *CoRR*.

Muennighoff, N. 2022. Sgpt: Gpt sentence embeddings for semantic search. *arXiv preprint arXiv:2202.08904*.

Muennighoff, N.; Rush, A. M.; Barak, B.; Scao, T. L.; Tazi, N.; Piktus, A.; Pyysalo, S.; Wolf, T.; and Raffel, C. 2023. Scaling Data-Constrained Language Models. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Rubin, O.; Herzig, J.; and Berant, J. 2022. Learning To Retrieve Prompts for In-Context Learning. In Carpuat, M.; de Marneffe, M.-C.; and Meza Ruiz, I. V., eds., *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2655–2671. Seattle, United States: Association for Computational Linguistics.

Sachdeva, N.; Coleman, B.; Kang, W.-C.; Ni, J.; Hong, L.; hsin Chi, E. H.; Caverlee, J.; McAuley, J. J.; and Cheng, D. Z. 2024. How to Train Data-Efficient LLMs. *ArXiv*, abs/2402.09668.

Silva, L.; and Barbosa, L. 2024. Improving dense retrieval models with LLM augmented data for dataset search. *Know.-Based Syst.*, 294(C).

Thrush, T.; Potts, C.; and Hashimoto, T. 2025. Improving Pretraining Data Using Perplexity Correlations. In *The Thirteenth International Conference on Learning Representations*.

Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; Rodriguez, A.; Joulin, A.; Grave, E.; and Lample, G. 2023. LLaMA: Open and Efficient Foundation Language Models. *ArXiv*, abs/2302.13971.

Wettig, A.; Gupta, A.; Malik, S.; and Chen, D. 2024. QuRating: Selecting High-Quality Data for Training Language Models. In *International Conference on Machine Learning (ICML)*.

Xia, M.; Malladi, S.; Gururangan, S.; Arora, S.; and Chen, D. 2024. LESS: Selecting Influential Data for Targeted Instruction Tuning. In *International Conference on Machine Learning (ICML)*.

Xie, S. M.; Santurkar, S.; Ma, T.; and Liang, P. 2023. Data Selection for Language Models via Importance Resampling. *Advances in Neural Information Processing Systems (NeurIPS)*.

Yang, L.; Zhang, Z.; Hong, S.; Xu, R.; Zhao, Y.; Shao, Y.; Zhang, W.; Yang, M.-H.; and Cui, B. 2022. Diffusion Models: A Comprehensive Survey of Methods and Applications. *ACM Computing Surveys*, 56: 1 – 39.

Yu, Z.; Das, S.; and Xiong, C. 2024. MATES: Model-Aware Data Selection for Efficient Pretraining with Data Influence Models. In *NeurIPS*.

Zhang, Y.; Li, Y.; Cui, L.; Cai, D.; Liu, L.; Fu, T.; Huang, X.; Zhao, E.; Zhang, Y.; Chen, Y.; et al. 2023. Siren's song in the AI ocean: a survey on hallucination in large language models. *arXiv preprint arXiv:2309.01219*.

Zhao, Y.; Du, L.; Ding, X.; Ouyang, Y.; Wang, H.; Xiong, K.; Gao, J.; Sun, Z.; Xu, D.; Yang, Q.; et al. 2025. Beyond Similarity: A Gradient-based Graph Method for Instruction Tuning Data Selection. *CoRR*.