# A Critical Look at Targeted Instruction Selection: Disentangling What Matters (and What Doesn't)

**Anonymous Authors**[1]

## Abstract

Instruction fine-tuning of large language models (LLMs) often involves selecting a subset of instruction training data from a large candidate pool, using a small query set from the target task. Despite growing interest, the literature on targeted instruction selection remains fragmented and opaque: methods vary widely in selection budgets, often omit zero-shot baselines, and frequently entangle the contributions of key components. As a result, practitioners lack actionable guidance on selecting instructions for their target tasks. In this work, we aim to bring clarity to this landscape by disentangling and systematically analyzing the two core ingredients: data representation and selection algorithms. Our framework enables controlled comparisons across models, tasks, and budgets. We find that only gradient-based data representations choose subsets whose similarity to the query consistently predicts performance across datasets and models. While no single method dominates, gradient-based representations paired with a greedy round-robin selection algorithm tend to perform best on average at low budgets, but these benefits diminish at larger budgets. Finally, we unify several existing selection algorithms as forms of approximate distance minimization between the selected subset and the query set, and support this view with new generalization bounds. More broadly, our findings provide critical insights and a foundation for more principled data selection in LLM fine-tuning.

## 1. Introduction

Large language models (LLMs) have demonstrated a remarkable ability to follow complex instructions through instruction fine-tuning on curated datasets of instruction-response pairs (Hurst et al., 2024; Olmo et al., 2025; Agarwal et al., 2025). Constructing such datasets, however, typically requires careful experimentation and extensive ablation studies, making them both time-consuming and computationally expensive (Guha et al., 2025; Magnusson et al., 2025). This challenge is further exacerbated by the growing need to adapt LLMs to specialized downstream tasks under limited data or compute budgets (Thulke et al., 2024; Zhao et al., 2024b). As a result, a growing body of work studies how to automatically select a subset of examples from a large candidate pool that is most useful for a given target task—a problem commonly referred to as *targeted instruction selection* (Xia et al., 2024; Liu et al., 2024b; Ivison et al., 2025; Yin & Rush, 2024).

Despite increasing interest, the literature on targeted instruction selection remains fragmented and difficult to interpret (Appendix A). Proposed methods vary widely in their formulations, often involve multiple design choices (e.g., representations, similarity metrics, selection algorithms), and lack key baselines, such as the zero-shot baseline. Empirical results are inconsistent, and it remains unclear which techniques actually drive performance, when, and why.

In this work, we bring clarity to this space through a disentangled framework that separates the two core ingredients of targeted instruction selection: (i) the representation used to encode fine-tuning data and (ii) the algorithm used to select examples based on these representations. This disentangled view allows us to isolate the effects of each component and enables controlled comparisons across models, datasets, and budgets. We further show that, despite differing in algorithmic detail, many selection methods can be unified under the view of approximate distance minimization between the selected subset and the target task distributions. We support this perspective with new generalization bounds that characterize when distance-based selection helps—and when it doesn't.

Our empirical findings are mixed but revealing. Gradient-based representations are the only ones whose similarity (or distance) to the query set reliably predicts performance (Section 5.1, Section 5.2), but even these fail to improve over zero-shot inference in some regimes (Appendix M).

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

While no single method dominates, greedy round-robin selection tends to perform best at small budgets, while optimal transport-based methods offer modest gains at larger ones (Section 5.2). Surprisingly, randomly sampled subsets often match or outperform many popular selection methods (TyDiQA and MMLU-Pro in Figure 5), especially as the budget increases, highlighting the brittleness of current practice.

Our work makes the following contributions:

- We propose a **disentangled framework** for targeted instruction selection that isolates the effects of (i) data representations and (ii) selection algorithms, enabling controlled comparisons.
- We conduct **systematic empirical analyses** across multiple target tasks and LLMs, showing that gradient-based data representations correlate most strongly with the loss and that greedy round-robin selection performs best at low budgets, while optimal transport-based methods perform best at high budgets.
- We develop a **unified theoretical perspective** that interprets many selection algorithms as approximate distance minimization, and prove generalization bounds that explain both the benefits and limitations of this view.

## 2. Targeted Instruction Selection

Let $\mathcal{D} = \{z_i = (x_i, y_i)\}_{i=1}^{N}$ be a large candidate pool and $\mathcal{Q} = \{q_j = (x_j, y_j)\}_{j=1}^{M}$ be the query set drawn from the distribution $P_{\mathcal{T}}$ of the target task $\mathcal{T}$. We consider a model $f_\theta$ parameterized by $\theta$. For a sample $z = (x, y)$, we define the per-example loss as $\ell(\theta; z) := \ell(f_\theta(x), y)$. The goal of targeted instruction selection is to choose a subset of examples $\mathcal{S} \subseteq \mathcal{D}$ of size $B$ (training budget) and train the model $f_\theta$ using only examples in $\mathcal{S}$ so as to minimize the expected loss on the target task $\mathcal{T}$:

$$\mathcal{S}^* = \arg \min_{\substack{\mathcal{S}\subseteq\mathcal{D} \\ |\mathcal{S}|=B}} \mathbb{E}_{z\sim P_{\mathcal{T}}}[\ell(\theta_{\mathcal{S}}; z)] .$$

Since we do not have access to the target tasks during training and selection, we use $\mathcal{Q}$ as a proxy for selection and choose $\mathcal{S}$ to solve the objective:

$$\hat{\mathcal{S}} = \arg \min_{\substack{\mathcal{S}\subseteq\mathcal{D} \\ |\mathcal{S}|=B}} \frac{1}{M} \sum_{j=1}^{M} \ell(\theta_{\mathcal{S}}; q_j) .$$

Optimizing this objective is impractical and combinatorially expensive, which has motivated compute efficient instruction selection methods.

## 3. A Disentangled View of Instruction Selection

In this section, we present the disentangled view of targeted instruction selection. Then, we describe the commonly used data representations and selection algorithms.


*Figure 1.* **Disentangled view of targeted instruction selection.** First, the query set (stars) and candidate pool (dots) are encoded as data representations. Then, for a given budget, using the data representations for the query and candidates, we perform targeted selection (denoted by the dotted line) using a selection algorithm such as greedy round-robin.

### 3.1. Disentangling Data Representation and Selection Algorithm

We adopt a disentangled view of instruction selection that separates two key components: (i) data representation, and (ii) the selection algorithm (Figure 1).

- **Data Representation**: First, we encode instruction-response pairs from the candidate pool and query sets into feature vectors (their *representations*). Ideally, these representations capture candidate-query distances that predict performance on the target task.
- **Selection Algorithm**: Next, the selection algorithm uses the data representations for the candidate pool and the query set to compute the similarity (or distances) between them and then select $B$ examples from the candidate pool based on the distances, where $B$ is the budget.

With this view, we aim to isolate the effects of data representations and selection algorithms from prior work that reports the best results on targeted instruction selection (Liu et al., 2024b; Xia et al., 2024; Ivison et al., 2025).

### 3.2. Data Representation

Here, we describe three data representation approaches for encoding the samples (More details in Appendix E). In this work, for a fair comparison, all these data representations are used to compute a cosine similarity matrix between the query and the candidate samples.

**RDS+.** We call the data representation in Ivison et al. (2025) as RDS+ and discuss the selection algorithm (RR) in Section 3.3. RDS+ computes the representations for the samples in the query and candidate pool by taking a position-weighted mean of the hidden states from the base language model (Muennighoff, 2022).

**EMBED.** Here, the query and candidate samples are passed through an off-the-shelf sentence encoder, such as GTR-T5 (Ni et al., 2022), to produce representations. Since these are typically much smaller, they significantly reduce FLOPs compared to RDS+. We follow the setup from Ivison et al. (2025) to produce these representations.

**LESS.** Xia et al. (2024) propose LESS (Low-rank gradiEnt Similarity Search), an optimization-aware influence formulation that represents query and candidate samples as low-dimensional gradient features. LESS estimates influence using a first-order approximation of training dynamics (Pruthi et al., 2020) and adapts it to Adam (Kingma & Ba, 2015). Concretely, it averages the cosine similarity between the query gradient and the candidate's Adam update vector across multiple training checkpoints. To make this approach scalable, LESS computes LoRA gradients (Hu et al., 2022) and applies random projection to obtain compact vectors, resulting in an efficient and reusable gradient datastore (Johnson & Lindenstrauss, 1984; Park et al., 2023). Recent variants build on the same principle to construct richer gradient-based representations (Wang et al., 2025). For convenience, we use LESS representations to encode the candidate pool and query sets (see Appendix D for details).

### 3.3. Selection Algorithm

We describe selection algorithms for targeted instruction selection (More details in Appendix F).

**Greedy Round-Robin (RR).** We consider the greedy round robin from Ivison et al. (2025). For each query sample, RR selects the sample from the candidate pool with the highest cosine similarity, adds the sample to the subset, and then removes it from the candidate pool. The algorithm repeats the round-robin process across all query samples until the data selection budget $B$ is exhausted.

**Doubly Greedy (DG).** We consider the doubly greedy selection algorithm from Xia et al. (2024). Given a similarity matrix, DG assigns each candidate sample an influence score equal to the maximum similarity it has with any query point. Then, DG selects the top-$B$ influential candidates to create the subset.

**KNN-Uniform.** KNN-Uniform is a selection algorithm inspired by optimal transport (Liu et al., 2024b). KNN-Uniform proposes a closed-form solution based on $K$ nearest neighbors to avoid explicitly solving the optimal transport problem. The algorithm first determines $K$ based on the trade-off between the alignment and diversity. After determining $K$, for each query, a uniform probability mass is assigned to the $K$ nearest neighbors, and then top-$B$ candidates with the highest mass (summed over queries) are selected.

**KNN-KDE.** KNN-KDE builds on the closed-form solution of KNN-Uniform by adding an additional regularizer to reduce the effect of near duplicates in the candidate pool (Liu et al., 2024b). Instead of assigning uniform mass to nearly identical points, Liu et al. (2024b) proposes incorporating kernel density estimation (KDE) as a regularizer. Then, for each query example, the $K$ nearest neighbors are assigned probability mass weighted by the inverse of their density estimates, and the top-$B$ candidates are selected.

**Unbalanced OT (UOT).** Here, we propose a new selection algorithm based on unbalanced optimal transport (Chizat et al., 2016), which we refer to as UOT. Unlike KNN-Uniform, in UOT, we explicitly solve the optimization problem of transporting mass from the query set to the candidate pool, penalizing marginal deviations (see Appendix C). This allows us to "ignore" outliers and less relevant samples for the target task. After solving for the transport plan, we sum the plan over the rows (queries) and choose the top-$B$ candidates with the highest mass (Appendix F.2 for implementation).

## 4. Experimental Setup

We closely follow the setup from Ivison et al. (2025) and Xia et al. (2024). Following Xia et al. (2024), we primarily use Llama-2-7B as the base model and train on the selected subset of training data. In Appendix M, we include experiments with additional models. Across all data representations, selection algorithms, and models, we use cosine similarity (or cosine distance) to measure the similarity between the query-candidate pairs. We experiment with the following target tasks: BBH, Codex, GSM8K, TyDiQA, and MMLU-Pro (Appendix B). We use the subsampled Tulu V2 dataset from Ivison et al. (2025) as the candidate pool. More training details and hyperparameters are included in Appendix G. We use the evaluation code from Ivison et al. (2025) and `lm-eval-harness` to evaluate the models and report the downstream metrics. We also report the average cross entropy loss over the response tokens on the query set (query loss).

## 5. Experiments and Analysis

### 5.1. Does Subset Distance to Query Predict Performance?

In this experiment, we aim to understand the relationship between the subset distance to the query and performance. The goal is to determine which data representation creates subsets whose distances predictably correlate with loss and downstream performance. If there is a correlation, we can formulate optimization objectives to minimize the distance between the query set and the candidates, thereby achieving the highest downstream performance. In Section 6, we also theoretically motivate this view by showing that minimizing the subset-query distance translates into minimizing the loss.

**Setup.** We consider all the data representation methods from Section 3.2. Given a cosine similarity matrix, we run the greedy round-robin selection algorithm over the

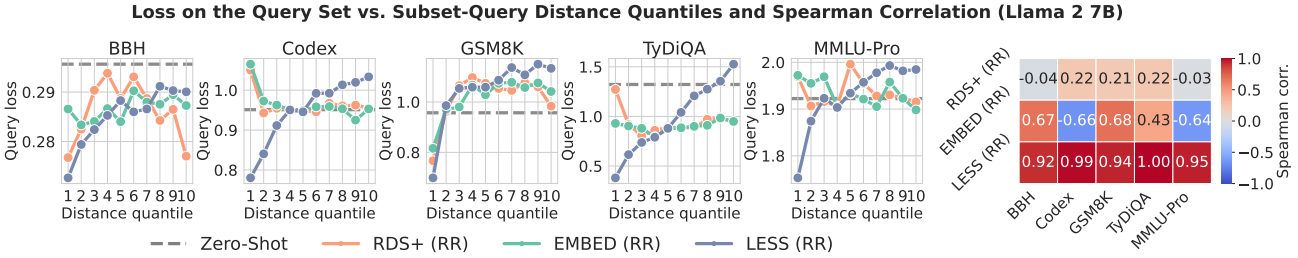


*Figure 2.* **Query loss vs. subset-query distance quantile.** We stratify candidates into 10 distance quantiles (1 = closest, 10 = farthest) using each representation, select 500 examples per quantile using the RR selection algorithm, and train the Llama 2 7B model. We report query-set cross-entropy loss and Spearman correlation per target task. LESS (RR) exhibits a strong monotonic increase in loss with distance (high positive Spearman correlation), whereas RDS+ (RR) and EMBED (RR) show weak or inconsistent correlations.

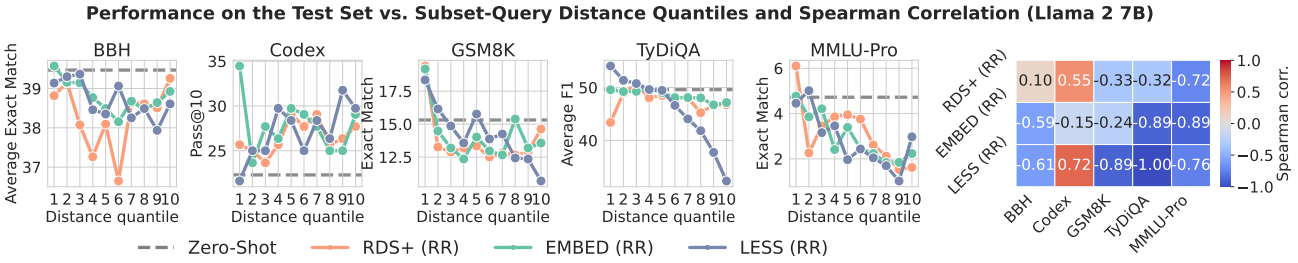


*Figure 3.* **Downstream performance vs. subset-query distance quantile.** Using the same quantile construction and training protocol as Figure 2, we evaluate downstream task performance across distance quantiles and report Spearman correlation per target task. LESS (RR) shows a strong negative correlation across most target tasks, while RDS+ (RR) and EMBED (RR) exhibit weaker, less consistent trends.

candidates to obtain an ordering, which we then use to stratify them into distance quantiles. The quantiles are indexed from 1 to 10, where 1 corresponds to the subset closest to the query and 10 to the farthest. Then, we select the top 500 training examples from each quantile and train Llama 2 7B on them.

**Results.** Figure 2 shows that only quantiles created by LESS (RR) highly correlate with the loss on the query set. On the other hand, both RDS+ (RR) and EMBED (RR) show very low Spearman correlation Across all target tasks, models trained on a subset in the first distance quantile using LESS (RR) achieve the lowest loss. In contrast, RDS+ and EMBED may not often show the lowest loss in their first quantile. To our surprise, we find that both methods can sometimes have the lowest loss in the last distance quantile (see BBH for RDS+ (RR) and MMLU-Pro for EMBED (RR)). Figure 3 shows that LESS (RR) shows a strong negative Spearman correlation (lower is better) across four out of the five downstream tasks. However, we do find that lower loss in Figure 2 does not necessarily translate to the highest downstream performance in the first quantile. For instance, we see that EMBED (RR) performs better than LESS (RR) on four out of the five target tasks despite having a much higher loss on the query set. We analyze the behavior of these representations by further dividing the first quantile in Appendix H.

Overall, LESS is the only representation that creates subsets whose distance correlates predictably with query loss and downstream performance. In Appendix M, across additional models, we find that LESS consistently shows strong correlation across target tasks, but the trends are less consistent on downstream performance with newer, over-trained models.

### 5.2. Effect of Data Representation across Subset Budgets

Here, we select subsets with different data representations while keeping the selection algorithm fixed to study performance as the budget increases.

**Setup.** We fixed the selection algorithm to a greedy round-robin approach, since Ivison et al. (2025) showed strong performance across tasks. Given a cosine similarity matrix from a data representation method, we select B samples from the candidate pool where $B \in \{500, 1000, 2500, 5000, 10000\}$. Then, we train the base model, Llama 2 7B, on the selected samples and report the downstream performance. We also include a Random baseline that uniformly samples $B$ candidates without replacement from the candidate pool. For all methods, we train three models with different seeds and report the average performance and standard error. For the Random baseline, we sample three times from the candidate pool, train one model on each, and report the average performance.

**Results.** Figure 4 shows that LESS (RR) achieves the lowest loss on the query set across all the target tasks under different budget constraints. We observe that RDS+ (RR)

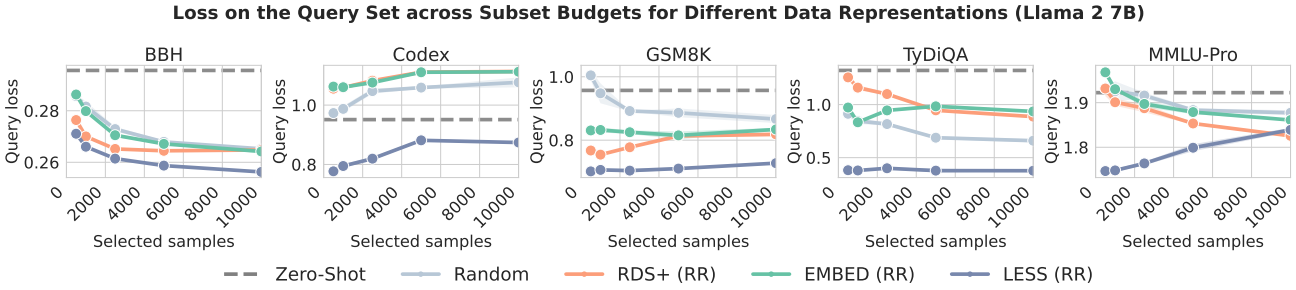


*Figure 4.* **Query loss vs. budget for different data representations (fixed selection algorithm).** Using greedy round-robin selection and the query-candidate pool similarity, we select subsets of size $B \in \{500, 1000, 2500, 5000, 10000\}$, train Llama 2 7B on them, and report average cross entropy loss averaged across three seeds and the standard error. Random averages over three uniformly sampled subsets from the candidate pool. LESS (RR) achieves the lowest loss across target tasks, while RDS+ (RR) and EMBED (RR) can underperform Random at larger budgets.

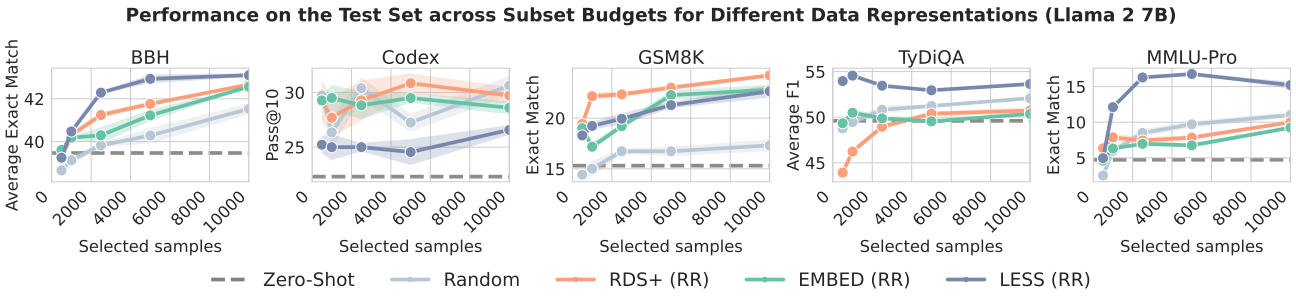


*Figure 5.* **Downstream performance vs. budget for different data representations (fixed selection algorithm).** With the same greedy round-robin selection and budgets as Figure 4, we report the downstream performance for different data representations averaged across three random seeds and the standard error. LESS (RR) performs best on BBH, TyDiQA, and MMLU-Pro, whereas RDS+ (RR) performs the best on GSM8K and is competitive with Random on Codex.

and EMBED (RR) can have higher loss than the Random baseline. Finally, we see that the query loss can increase or may stabilize as we continue increasing the budget. Figure 5 shows that LESS (RR) outperforms the other methods by a clear margin on three out of the five target tasks. Next, we observe that RDS+ (RR) achieves higher performance across all budget constraints on GSM8K, but Random is a competitive baseline at larger budgets on Codex. Lastly, we note that Random is a strong baseline but often underperforms targeted instruction selection methods at low budgets.

Overall, these results show that, under a fixed selection algorithm (greedy round-robin), gradient-based representations (LESS) perform better than model-based representations on a majority of the target tasks. Across models and downstream target tasks that benefit from additional training (i.e., zero-shot performance is not saturated), we see that LESS often outperforms other baselines (Appendix M). However, a key limitation of LESS representations is that they are computationally more expensive than model-based representations, as they require a forward and a backward pass over the entire candidate pool. For this reason, in Appendix J, we further investigate whether subsets created by smaller proxy LESS models offer a cheaper alternative, and show that even models with 135M parameters can select examples for training larger models and dramatically reduce FLOPs. Finally, we find that subsets selected by model-based representations are competitive and often outperform Random under constrained budgets.

### 5.3. Effect of Selection Algorithms across Subset Budgets

Here, we select subsets with different selection algorithms while keeping the data representation fixed to study the performance as the budget increases.

**Setup.** The experimental setup closely follows Section 5.2. Building on our insights from Section 5.2, we fix LESS as our data representation. Given the LESS representations for the query set and candidate pool, we use the selection algorithm to select samples for budgets $B \in \{500, 1000, 2500, 5000, 10000\}$. We use the weighted cosine similarity matrix between the query and the candidate pool (Equation 2 in the Appendix) to select instructions using greedy round-robin (RR) and doubly greedy (DG). We convert this cosine similarity matrix into a cosine distance matrix to select samples with UOT, KNN-Uniform, and KNN-KDE (Appendix F.1 and F.2). Following Liu et al. (2024b), we also include results for KNN-Uniform and KNN-KDE using L2 distance in Appendix K.

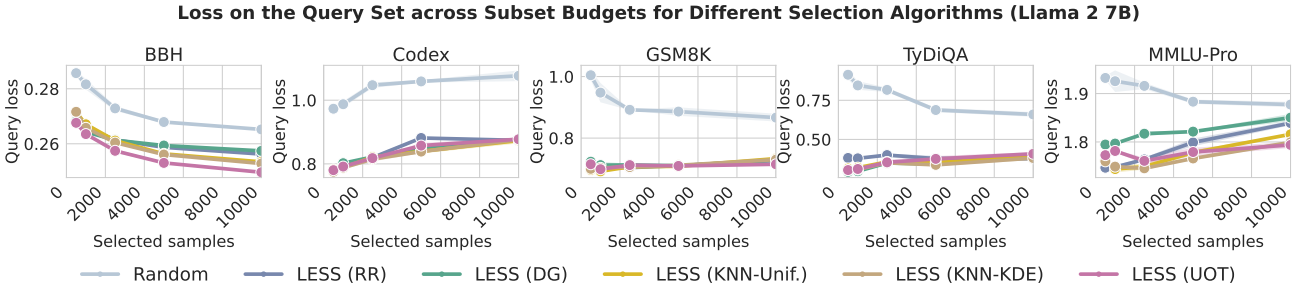


*Figure 6.* **Query loss vs. budget for different selection algorithms (fixed data representation).** Using LESS representations and the query-candidate pool cosine similarity (or distance), we select subsets of size $B \in \{500, 1000, 2500, 5000, 10000\}$ with each selection algorithm, train Llama 2 7B on them, and report average cross entropy loss on the query set averaged across three seeds and the standard error. Random averages over three uniformly sampled subsets from the candidate pool. UOT achieves the lowest loss on three of the five datasets and remains competitive on the others, while DG often underperforms, yielding the highest loss on three datasets.

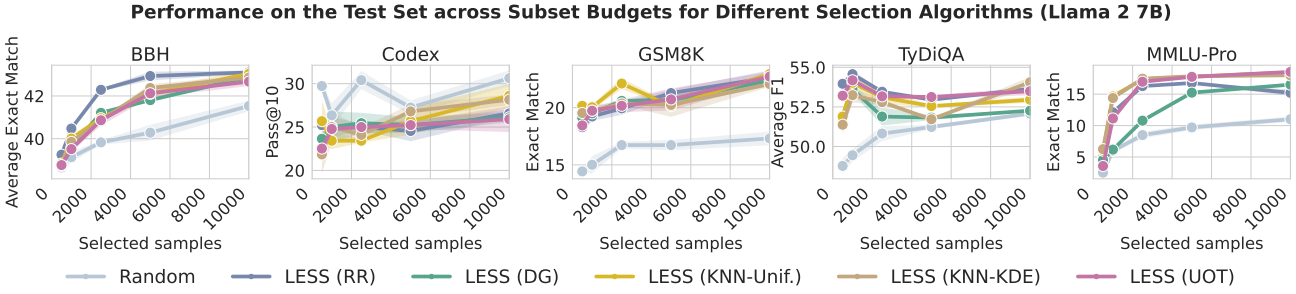


*Figure 7.* **Downstream performance vs. budget for different selection algorithms (fixed data representation).** With the same data representation and the budgets as Figure 6, we report downstream performance for different selection algorithms averaged across three seeds and the standard error. RR tends to perform best at smaller budgets, whereas UOT and KNN-KDE perform better at larger budgets; DG consistently underperforms across three of the five datasets.

**Results.** Figure 6 shows that across the five tasks, UOT achieves the lowest loss on three of them, while remaining competitive on the others. On the other hand, we see that DG often has the highest loss across tasks at large budgets. Figure 7 shows the downstream performance after training Llama-2-7B on the selected subsets. We observe that LESS (RR), the greedy round-robin selection algorithm, performs well on low budgets. On the other hand, LESS (UOT) and LESS (KNN-KDE) perform better with higher budgets. Finally, we find that the doubly greedy (DG) algorithm consistently underperforms other selection algorithms, suggesting that it might be choosing samples closest to only a subset of the queries.

Overall, our results show that greedy round-robin (RR) selection works best under limited budget constraints, but these gains generally diminish as more examples are added. At larger budgets, optimal transport–based selection methods (UOT and KNN-KDE) provide greater benefits. In Appendix M, we find that the performance trends observed in this experiment generalize to other models. For example, LESS (RR) performs best on BBH with Llama 3.2 3B and Qwen3 4B Base. We also observe that LESS (UOT) performs best on MMLU-Pro at high budgets with SmolLM3 and Olmo3 7B Base. These results suggest that selection algorithms exhibit similar performance trends across models for particular target tasks.

## 6. A Unifying View: Instruction Selection as Set Distance Minimization

While subset selection algorithms vary in form, from greedy heuristics to transport-based selection, they often share a deeper, unifying principle: selecting a subset that is "close" to the query set in the representation space. In this section, we formalize and motivate this perspective.

Concretely, we observe that a broad class of existing methods (including those discussed in Section 3.3) can be viewed as approximately minimizing a distributional distance between the selected subset $\mathcal{S}$ and the query set $\mathcal{Q}$:

- Greedy algorithms (e.g., round-robin, doubly greedy) aim to reduce the average or maximum similarity-based distance between $\mathcal{S}$ and $\mathcal{Q}$.
- Density-based methods (e.g., KNN-Uniform, KNN-KDE) select points from high-density regions near $\mathcal{Q}$, implicitly matching the mass of the subset distribution to the query.
- Optimal transport approaches (e.g., UOT) directly solve for a transport coupling minimizing cost between empirical distributions associated with $\mathcal{S}$ and $\mathcal{Q}$.

Despite differing in implementation and exact objective, all these methods can be understood as minimizing a distance function $\mathrm{Dist}(\mathcal{S}, \mathcal{Q})$, where the choice of distance reflects the algorithm's inductive biases. This unified view not only clarifies relationships between methods, but also explains their performance patterns. In the remainder of this section:

- We show that reducing the subset-query distance tightens a generalization bound on downstream loss (Section 6.1).
- We show that the benefit of distance-aware selection over random sampling diminishes with budget, and characterize this tradeoff (Section 6.2).

### 6.1. Minimizing Subset-Query Distance Tightens a Generalization Bound

We establish a theoretical bound formalizing the intuition that selecting a subset $\mathcal{S}$ whose empirical distribution is close to that of the query set $\mathcal{Q}$ leads to improved performance on the target task $\mathcal{T}$. We show that the empirical test loss$L_{\mathcal{T}}(\theta_S) := \frac{1}{|\mathcal{T}|}\sum_{z\in\mathcal{T}} \ell(\theta_S; z)$ for a loss function $\ell(\theta_S; z)$ and $\theta_{\mathcal{S}} \in \arg\min_{\theta\in\Theta} L_{\mathcal{S}}(\theta)$ an empirical risk minimizer trained on $\mathcal{S}$, is upper-bounded by the 1-Wasserstein distance between the empirical distributions of the selected subset and the query set, denoted by $W_1(\hat{P}_{\mathcal{S}}, \hat{P}_{\mathcal{Q}})$. Here, $\hat{P}_{\mathcal{S}}$ and $\hat{P}_{\mathcal{Q}}$ denote the empirical distributions associated with $\mathcal{S}$ and $\mathcal{Q}$, respectively. This result follows by viewing subset selection as a two-stage domain adaptation problem: first from $\mathcal{S}$ (source) to $\mathcal{Q}$ (target), and then from $\mathcal{Q}$ (source), to $\mathcal{T}$ (target). Bounds of this form are well established in the domain adaptation literature, where optimal transport distances between source and target distributions are known to control transfer error (Redko et al., 2017; Courty et al., 2017). In our setting, this bound implies that algorithms which approximately minimize $W_1(\hat{P}_{\mathcal{S}}, \hat{P}_{\mathcal{Q}})$ directly tighten a theoretical upper bound on target task performance.

**Theorem 6.1.** *Let $\ell : \Theta \times \mathcal{Z} \to \mathbb{R}_+$ be a loss function, where $\mathcal{Z} \subset \mathbb{R}^d$ denotes the data space and $\Theta$ the parameter space. Assume that $\ell$ is symmetric, convex, bounded, satisfies the triangle inequality, and for $z = (x, y) \in \mathcal{Z}$ admits the parametric form $\ell(\theta; z) = |y - f_\theta(x)|^q$ for some $q > 0$. Let $\mathcal{D}$ denote a labeled candidate pool, and let $\mathcal{S} \subseteq \mathcal{D}$ be any subset of size $B := |\mathcal{S}|$. Let $\mathcal{Q}$ (the query set) and $\mathcal{T}$ (the test set) be labeled datasets, and assume $|Q| \le \min(|\mathcal{T}|, |\mathcal{S}|)$. Then for any $d' > d$ and $c' < \sqrt{2}$, there exists a constant $N_0$, depending on $d'$, such that for any $\delta > 0$ and $|Q| \ge N_0 \max(\delta^{-(d'+2)}, 1)$, with probability at least $1 - 2\delta$:*

$$L_{\mathcal{T}}(\theta_{\mathcal{S}}) \le \underbrace{W_1(\hat{P}_{\mathcal{S}}, \hat{P}_{\mathcal{Q}})}_{\substack{\text{Subset and query}\\ \text{dataset distance}}} + \underbrace{W_1(\hat{P}_{\mathcal{Q}}, \hat{P}_{\mathcal{T}})}_{\substack{\text{Query and test}\\ \text{dataset distance}}} + \underbrace{L_{\mathcal{S}}(\theta_{\mathcal{S}})}_{\text{training error}} + \zeta\sqrt{\frac{2}{c'}\log\left(\frac{1}{\delta}\right)} + \tilde{\lambda} \tag{1}$$

*where $W_1$ is the 1-Wasserstein distance, $\zeta$ is a constant given by $\zeta := B^{-\frac{1}{2}} + 2|\mathcal{Q}|^{-\frac{1}{2}} + |\mathcal{T}|^{-\frac{1}{2}}$, and $\tilde{\lambda}$ is the minimum combined error $L_{\mathcal{S}}(\theta_{\tilde{\mathcal{S}}}) + 2L_{\mathcal{Q}}(\theta_{\tilde{\mathcal{S}}}) + L_{\mathcal{T}}(\theta_{\tilde{\mathcal{S}}})$ over datasets $\tilde{\mathcal{S}} \subseteq \mathcal{D}$ of size $B$.*

*Proof in Appendix L.2*

Among the terms in equation 1, only $W_1(\hat{P}_{\mathcal{S}}, \hat{P}_{\mathcal{Q}})$ is directly affected by the choice of $\mathcal{S}$ (for fixed $\mathcal{Q}, \mathcal{T}$). Consequently, selecting $\mathcal{S}$ to (approximately) minimize $W_1(\hat{P}_{\mathcal{S}}, \hat{P}_{\mathcal{Q}})$ is a principled objective: any reduction in this transport distance directly tightens the right-hand side, and hence the worst-case upper bound on $L_{\mathcal{T}}(\theta_{\mathcal{S}})$. The second Wasserstein term, $W_1(\hat{P}_{\mathcal{Q}}, \hat{P}_{\mathcal{T}})$, quantifies mismatch between the query distribution and the downstream evaluation distribution. This term is independent of $\mathcal{S}$, and therefore sets an irreducible error on how informative $\mathcal{Q}$ is about $\mathcal{T}$.

At the same time, the theorem does not imply that minimizing $W_1(\hat{P}_{\mathcal{S}}, \hat{P}_{\mathcal{Q}})$ alone guarantees strong target performance. The bound also includes the training error $L_{\mathcal{S}}(\theta_{\mathcal{S}})$ and the term $\tilde{\lambda}$. In the instruction-selection setting, these terms capture two additional failure modes: (i) $\mathcal{S}$ may be close to $\mathcal{Q}$ yet noisy or internally inconsistent, leading to large $L_{\mathcal{S}}(\theta_{\mathcal{S}})$; and (ii) there may be no single hypothesis that performs well simultaneously on $\mathcal{S}$, $\mathcal{Q}$, and $\mathcal{T}$, leading to large $\tilde{\lambda}$. The quantity $\tilde{\lambda}$ can be interpreted as an ideal joint error term: it is small when the candidate pool $\mathcal{D}$ contains a size-$B$ subset whose ERM achieves low loss on $\mathcal{S}$, $\mathcal{Q}$, and $\mathcal{T}$ simultaneously. Consequently, if $\mathcal{D}$ lacks coverage of the skills required by $\mathcal{Q}$ and $\mathcal{T}$, or if the hypothesis class cannot realize a predictor that works well across these datasets, then no selection rule based purely on distribution matching can guarantee strong downstream performance. The remaining terms account for finite-sample effects through $\zeta$.

### 6.2. Diminishing Returns of Query-Aware Selection as Budget Increases

Computing the exact subset that minimizes the distance to the query set is a combinatorial problem, and even approximate distance-minimization methods can be computationally expensive. In contrast, uniformly sampling a subset from the candidate pool $\mathcal{D}$ is essentially cost-free and becomes increasingly competitive as the selection budget $B$ grows (in the extreme, when $B \approx |\mathcal{D}|$, random and optimized subsets nearly coincide). For a fixed budget $B$, Theorem 6.2 formalizes the benefit of query-aware selection by upper bounding the improvement in test loss achieved by training on a Wasserstein-optimal subset—i.e., the subset closest to $\mathcal{Q}$—relative to a uniformly random subset of the same size.

**Theorem 6.2.** *Let the candidate pool $\mathcal{D} \subset \mathbb{R}^d$ lie in a set of diameter $\Delta$ and $d \ge 3$. Let $\mathcal{S}^{\mathrm{rnd}} \subseteq \mathcal{D}$ be a subset of size $B$ sampled uniformly at random, and let $S^*_W \subseteq \mathcal{D}$ be a subset of size $B$ that minimizes the 1-Wasserstein distance to the query set $\mathcal{Q}$, i.e., $\mathcal{S}^*_W \in \arg\min_{\mathcal{S}\subseteq\mathcal{D};|\mathcal{S}|=B} W_1(\hat{P}_{\mathcal{S}}, \hat{P}_{\mathcal{Q}})$.*

*Assume (i) $L_{\mathcal{S}}(\theta)$ is $\mu$–strongly convex in $\theta$, (ii) $L_{\mathcal{T}}(\theta)$ is $K$-Lipschitz, and (iii) $\nabla_\theta \ell(\theta; z)$ is $G_{\theta z}$-Lipschitz with respect to $z$. Then, there exists a constant $C_d > 0$, depending only on the dimension $d$, such that, with probability at least $1 - 2\delta$:*

$$L_{\mathcal{T}}(\theta_{\mathcal{S}^{\mathrm{rnd}}}) - L_{\mathcal{T}}(\theta_{S^*_W}) \leq C_\star \Bigg( \underbrace{C_d \Delta B^{-1/d}}_{\substack{\text{Curse of} \\ \text{dimensionality}}} + \underbrace{\Delta \sqrt{\frac{2\log(1/\delta)}{B}}}_{\text{Concentration bound}} + \underbrace{W_1(\hat{P}_{\mathcal{D}}, \hat{P}_{\mathcal{Q}})}_{\text{Pool-query mismatch}} + \underbrace{W_1(\hat{P}_{S^*_W}, \hat{P}_{\mathcal{Q}})}_{\text{Distance residual}} \Bigg)$$

*Proof in Appendix L.3.*

Theorem 6.2 shows that the potential improvement in test loss from training on a query-aware subset $S^*_W$ (rather than a uniformly random subset $\mathcal{S}^{\mathrm{rnd}}$ of the same size) decreases as the budget $B$ increases. In particular, when $d \geq 3$, the leading $B$-dependent term in the upper bound scales as $C_\star C_d \Delta\, B^{-1/d}$ (the $B^{-1/2}$ concentration term is lower order). Thus, up to the additive shift induced by pool–query mismatch $W_1(\hat{P}_{\mathcal{D}}, \hat{P}_{\mathcal{Q}})$ (and the residual $W_1(\hat{P}_{S^*_W}, \hat{P}_{\mathcal{Q}})$), we can define a critical subset size for a tolerance $\varepsilon$: requiring $C_\star C_d \Delta\, B^{-1/d} \leq \varepsilon$ yields $B \geq (C_\star C_d \Delta/\varepsilon)^d$. This highlights the curse of dimensionality: in high-dimensional embedding spaces, making random sampling competitive can require $B$ to grow on the order of $\varepsilon^{-d}$, whereas query-aware selection can achieve a smaller loss gap at substantially smaller budgets.

Figure 8 shows how the performance gap between LESS variants and uniformly random subsets changes as the budget increases. As $B$ increases, the gap shrinks, and the performance approaches the random sampling, which is qualitatively aligned with Theorem 6.2. While $B^{-1/d}$ reference rate shows a slow worst-case decay in high dimensions, we see that LESS (KNN-Unif.), LESS (KNN-KDE), and LESS (UOT) exhibit a slower or a similar decay up to 2500 samples, suggesting the bound is non-trivial in this regime. Finally, the LESS variants approach the random baseline at different rates, suggesting that the choice selection algorithm affects the constants and residual errors.

## 7. Related Work

**Subset Selection.** Subset selection (or coreset selection) is a fundamental task in machine learning, where the goal is to choose a subset of the data from a large candidate pool for training and achieve downstream performance similar to or better than that of training on the entire candidate pool (Wei et al., 2015; Huang et al., 2019). Over the years, several techniques involving clustering (Har-Peled & Mazumdar, 2004; Chen et al., 2023), gradient matching (Killamsetty et al., 2021), kernel thinning (Dwivedi & Mackey, 2024; Carrell et al., 2025), and proxy models (Ye et al., 2024; Magnusson et al., 2025) have been proposed to efficiently select core training samples in a variety of domains (Hulkund et al., 2025). In this work, we focus on choosing a subset of examples from a candidate pool of instruction-response pairs for a target downstream task.

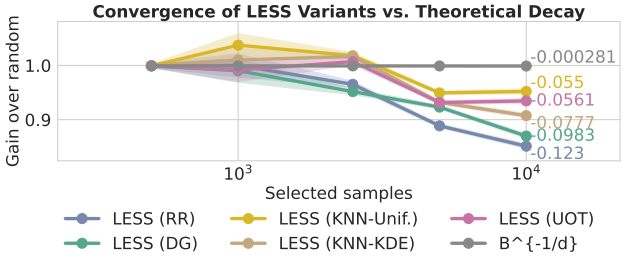


*Figure 8.* **Convergence of LESS variants toward random sampling as budget increases.** We plot the (normalized) loss gap between random sampling and each of the LESS variants as a function of the subset size $B$ (log-scale). The gray reference line indicates the $B^{-1/d}$ decay predicted by Theorem 6.2 (up to constants and additive mismatch/residual terms). For visual comparison of decay rates, we apply a constant offset to each LESS curve so that all methods start at $B_0^{-1/d}$ where $B_0 = 500$ and $d = 8192$; this shift does not affect slopes.

**Instruction Selection.** Instruction selection is a key ingredient in today's large language model training pipeline (Olmo et al., 2025). While curation via careful experimentation in the post-training pipeline has led to dramatic performance improvements (Longpre et al., 2023; Guha et al., 2025), there is a growing interest in automatically curating instruction tuning datasets (Liu et al., 2024a). We focus on instruction selection for target tasks where we select training examples under a given budget from a candidate pool. Existing work on automatic instruction selection often relies on heuristics such as length (Zhao et al., 2024a), perplexity (Ankner et al., 2024), number of reasoning (Li et al., 2025), task similarity (Xia et al., 2024; Ivison et al., 2025), and more (Albalak et al., 2024). Our focus is to understand targeted instruction selection that uses similarity between the query and the candidate pool to select subsets.

## 8. Conclusion

We disentangle the key components of targeted instruction selection and provide new critical empirical and theoretical insights. Our experiments reveal that gradient-based data representations create subsets whose distances to the query strongly predict performance. Exploring computationally cheaper alternatives to produce such representations is a promising research direction (Appendix J). Our theoretical insight, which unifies several existing selection algorithms as approximate distance minimizers, provides a general framework for creating new selection algorithms. Overall, we present a practical roadmap for designing improved targeted instruction selection frameworks.

## Impact Statement

This paper presents work whose goal is to advance targeted instruction selection in large language models (LLMs). While our work comprehensively evaluates numerous LLMs across many target tasks, we rely on existing pre-trained LLMs. Further fine-tuning on instructions might amplify any pre-existing biases in these pre-trained LLMs or might even hurt performance on unrelated tasks. Before deploying any of these models, we suggest carrying out careful evaluations on target tasks and additional safety evaluations.

## References


Agarwal, S., Ahmad, L., Ai, J., Altman, S., Applebaum, A., Arbus, E., Arora, R. K., Bai, Y., Baker, B., Bao, H., et al. gpt-oss-120b & gpt-oss-20b model card. *ArXiv preprint*, abs/2508.10925, 2025. URL https://arxiv.org/abs/2508.10925.

Albalak, A., Elazar, Y., Xie, S. M., Longpre, S., Lambert, N., Wang, X., Muennighoff, N., Hou, B., Pan, L., Jeong, H., Raffel, C., Chang, S., Hashimoto, T., and Wang, W. Y. A survey on data selection for language models. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=XfHWcNTSHp. Survey Certification, Featured Certification.

Allal, L. B., Lozhkov, A., Bakouch, E., von Werra, L., and Wolf, T. Smollm - blazingly fast and remarkably powerful, 2024.

Allal, L. B., Lozhkov, A., Bakouch, E., Blázquez, G. M., Penedo, G., Tunstall, L., Marafioti, A., Kydlíček, H., Lajarín, A. P., Srivastav, V., et al. Smollm2: When smol goes big–data-centric training of a small language model. *arXiv preprint arXiv:2502.02737*, 2025.

Alvarez-Melis, D. and Fusi, N. Geometric dataset distances via optimal transport. *ArXiv*, abs/2002.02923, 2020. URL https://api.semanticscholar.org/CorpusID:211066128.

Ankner, Z., Blakeney, C., Sreenivasan, K., Marion, M., Leavitt, M. L., and Paul, M. Perplexed by perplexity: Perplexity-based data pruning with small reference models. *ArXiv preprint*, abs/2405.20541, 2024. URL https://arxiv.org/abs/2405.20541.

Arjovsky, M., Chintala, S., and Bottou, L. Wasserstein gan. *ArXiv*, abs/1701.07875, 2017. URL https://api.semanticscholar.org/CorpusID:13943041.

Bakouch, E., Ben Allal, L., Lozhkov, A., Tazi, N., Tunstall, L., Patiño, C. M., Beeching, E., Roucher, A., Reedi, A. J., Gallouédec, Q., Rasul, K., Habib, N., Fourrier, C., Kydlicek, H., Penedo, G., Larcher, H., Morlon, M., Srivastav, V., Lochner, J., Nguyen, X.-S., Raffel, C., von Werra, L., and Wolf, T. SmolLM3: smol, multilingual, long-context reasoner. https://huggingface.co/blog/smollm3, 2025.

Benamou, J.-D. Numerical resolution of an "unbalanced" mass transport problem. *ESAIM: Mathematical Modelling and Numerical Analysis*, 37(5):851–868, 2003.

Biderman, S., Schoelkopf, H., Anthony, Q. G., Bradley, H., O'Brien, K., Hallahan, E., Khan, M. A., Purohit, S., Prashanth, U. S., Raff, E., et al. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pp. 2397–2430. PMLR, 2023.

Carrell, A. M., Gong, A., Shetty, A., Dwivedi, R., and Mackey, L. Low-rank thinning. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=iAkg2nVmvN.

Chen, H., Zhang, Y., Zhang, Q., Yang, H., Hu, X., Ma, X., Yanggong, Y., and Zhao, J. Maybe only 0.5% data is needed: A preliminary exploration of low training data instruction tuning. *ArXiv preprint*, abs/2305.09246, 2023. URL https://arxiv.org/abs/2305.09246.

Chen, M., Tworek, J., Jun, H., Yuan, Q., Pondé, H., Kaplan, J., Edwards, H., Burda, Y., Joseph, N., Brockman, G., Ray, A., Puri, R., Krueger, G., Petrov, M., Khlaaf, H., Sastry, G., Mishkin, P., Chan, B., Gray, S., Ryder, N., Pavlov, M., Power, A., Kaiser, L., Bavarian, M., Winter, C., Tillet, P., Such, F. P., Cummings, D. W., Plappert, M., Chantzis, F., Barnes, E., Herbert-Voss, A., Guss, W. H., Nichol, A., Babuschkin, I., Balaji, S., Jain, S., Carr, A., Leike, J., Achiam, J., Misra, V., Morikawa, E., Radford, A., Knight, M. M., Brundage, M., Murati, M., Mayer, K., Welinder, P., McGrew, B., Amodei, D., McCandlish, S., Sutskever, I., and Zaremba, W. Evaluating large language models trained on code. *ArXiv*, abs/2107.03374, 2021. URL https://api.semanticscholar.org/CorpusID:235755472.

Chizat, L., Peyr'e, G., Schmitzer, B., and Vialard, F.-X. Scaling algorithms for unbalanced transport problems. *arXiv: Optimization and Control*, 2016. URL https://api.semanticscholar.org/CorpusID:119312616.

Courty, N., Flamary, R., Tuia, D., and Rakotomamonjy, A. Optimal transport for domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39:1853–1865, 2014. URL https://api.semanticscholar.org/CorpusID:13347901.

Courty, N., Flamary, R., Habrard, A., and Rakotomamonjy, A. Joint distribution optimal transportation for domain adaptation. *Advances in neural information processing systems*, 30, 2017.

Dwivedi, R. and Mackey, L. Kernel thinning. *Journal of Machine Learning Research*, 25(152):1–77, 2024. URL http://jmlr.org/papers/v25/21-1334.html.

Flamary, R., Courty, N., Gramfort, A., Alaya, M. Z., Boisbunon, A., Chambon, S., Chapel, L., Corenflos, A., Fatras, K., Fournier, N., Gautheron, L., Gayraud, N. T., Janati, H., Rakotomamonjy, A., Redko, I., Rolet, A., Schutz, A., Seguy, V., Sutherland, D. J., Tavenard, R., Tong, A., and Vayer, T. Pot: Python optimal transport. *Journal of Machine Learning Research*, 22(78):1–8, 2021. URL http://jmlr.org/papers/v22/20-451.html.

Flamary, R., Vincent-Cuaz, C., Courty, N., Gramfort, A., Kachaiev, O., Quang Tran, H., David, L., Bonet, C., Cassereau, N., Gnassounou, T., Tanguy, E., Delon, J., Collas, A., Mazelet, S., Chapel, L., Kerdoncuff, T., Yu, X., Feickert, M., Krzakala, P., Liu, T., and Fernandes Montesuma, E. Pot python optimal transport (version 0.9.5), 2024. URL https://github.com/PythonOT/POT.

Fournier, N. and Guillin, A. On the rate of convergence in wasserstein distance of the empirical measure. *Probability theory and related fields*, 162(3):707–738, 2015.

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Guha, E., Marten, R., Keh, S., Raoof, N., Smyrnis, G., Bansal, H., Nezhurina, M., Mercat, J., Vu, T., Sprague, Z., et al. Openthoughts: Data recipes for reasoning models. *ArXiv preprint*, abs/2506.04178, 2025. URL https://arxiv.org/abs/2506.04178.

Har-Peled, S. and Mazumdar, S. On coresets for k-means and k-median clustering. In *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pp. 291–300, 2004.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Huang, L., Jiang, S., and Vishnoi, N. Coresets for clustering with fairness constraints. *Advances in neural information processing systems*, 32, 2019.

Hulkund, N., Maalouf, A., Cai, L., Yang, D., Wang, T.-H., O'Neil, A., Haucke, T., Mukherjee, S., Ramaswamy, V., Shen, J. H., et al. Datasˆ 3: Dataset subset selection for specialization. *ArXiv preprint*, abs/2504.16277, 2025. URL https://arxiv.org/abs/2504.16277.

Hurst, A., Lerer, A., Goucher, A. P., Perelman, A., Ramesh, A., Clark, A., Ostrow, A., Welihinda, A., Hayes, A., Radford, A., et al. Gpt-4o system card. *ArXiv preprint*, abs/2410.21276, 2024. URL https://arxiv.org/abs/2410.21276.

Ivison, H., Wang, Y., Pyatkin, V., Lambert, N., Peters, M., Dasigi, P., Jang, J., Wadden, D., Smith, N. A., Beltagy, I., et al. Camels in a changing climate: Enhancing lm adaptation with tulu 2. *ArXiv preprint*, abs/2311.10702, 2023. URL https://arxiv.org/abs/2311.10702.

Ivison, H., Zhang, M., Brahman, F., Koh, P. W., and Dasigi, P. Large-Scale Data Selection for Instruction Tuning. *ArXiv preprint*, abs/2503.01807, 2025. URL https://arxiv.org/abs/2503.01807.

Johnson, W. B. and Lindenstrauss, J. Extensions of lipschitz mappings into hilbert space. *Contemporary mathematics*, 26:189–206, 1984. URL https://api.semanticscholar.org/CorpusID:117819162.

Khaddaj, A., Engstrom, L., and Madry, A. Small-to-large generalization: Training data influences models consistently across scale. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=79ZkWgY2FI.

Killamsetty, K., Sivasubramanian, D., Ramakrishnan, G., De, A., and Iyer, R. K. GRAD-MATCH: gradient matching based data subset selection for efficient deep model training. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 5464–5474. PMLR, 2021. URL http://proceedings.mlr.press/v139/killamsetty21a.html.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In Bengio, Y. and LeCun, Y. (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980.

Li, Y., Emad, Y., Padthe, K., Lanchantin, J., Yuan, W., Nguyen, T., Weston, J., Li, S.-W., Wang, D., Kulikov, I., et al. Naturalthoughts: Selecting and distilling reasoning traces for general reasoning tasks. *ArXiv preprint*, abs/2507.01921, 2025. URL https://arxiv.org/abs/2507.01921.

Liero, M., Mielke, A., and Savaré, G. Optimal entropy-transport problems and a new hellinger–kantorovich distance between positive measures. *Inventiones mathematicae*, 211(3):969–1117, 2018.

Liu, W., Zeng, W., He, K., Jiang, Y., and He, J. What makes good data for alignment? A comprehensive study of automatic data selection in instruction tuning. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024a. URL https://openreview.net/forum?id=BTKAeLqLMw.

Liu, Z., Karbasi, A., and Rekatsinas, T. TSDS: data selection for task-specific model finetuning. In Globersons, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J. M., and Zhang, C. (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024b. URL http://papers.nips.cc/paper_files/paper/2024/hash/13848b5893119ff772b69812c95914fa-Abstract-Conference.html.

Longpre, S., Hou, L., Vu, T., Webson, A., Chung, H. W., Tay, Y., Zhou, D., Le, Q. V., Zoph, B., Wei, J., and Roberts, A. The flan collection: Designing data and methods for effective instruction tuning. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 22631–22648. PMLR, 2023. URL https://proceedings.mlr.press/v202/longpre23a.html.

Magnusson, I., Tai, N., Bogin, B., Heineman, D., Hwang, J. D., Soldaini, L., Bhagia, A., Liu, J., Groeneveld, D., Tafjord, O., et al. Datadecide: How to predict best pretraining data with small experiments. *ArXiv preprint*, abs/2504.11393, 2025. URL https://arxiv.org/abs/2504.11393.

McDiarmid, C. *On the method of bounded differences*, pp. 148–188. London Mathematical Society Lecture Note Series. Cambridge University Press, 1989.

Muennighoff, N. Sgpt: Gpt sentence embeddings for semantic search. *ArXiv preprint*, abs/2202.08904, 2022. URL https://arxiv.org/abs/2202.08904.

Ni, J., Qu, C., Lu, J., Dai, Z., Hernandez Abrego, G., Ma, J., Zhao, V., Luan, Y., Hall, K., Chang, M.-W., and Yang, Y. Large dual encoders are generalizable retrievers. In Goldberg, Y., Kozareva, Z., and Zhang, Y. (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 9844–9855, Abu Dhabi, United Arab Emirates, 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.669. URL https://aclanthology.org/2022.emnlp-main.669.

Olmo, T., Ettinger, A., Bertsch, A., Kuehl, B., Graham, D., Heineman, D., Groeneveld, D., Brahman, F., Timbers, F., Ivison, H., et al. Olmo 3. *ArXiv preprint*, abs/2512.13961, 2025. URL https://arxiv.org/abs/2512.13961.

Park, S. M., Georgiev, K., Ilyas, A., Leclerc, G., and Madry, A. TRAK: attributing model behavior at scale. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 27074–27113. PMLR, 2023. URL https://proceedings.mlr.press/v202/park23c.html.

Peyré, G. and Cuturi, M. Computational optimal transport. *Found. Trends Mach. Learn.*, 11:355–607, 2018. URL https://api.semanticscholar.org/CorpusID:73725148.

Pruthi, G., Liu, F., Kale, S., and Sundararajan, M. Estimating training data influence by tracing gradient descent. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/e6385d39ec9394f2f3a354d9d2b88eec-Abstract.html.

Redko, I., Habrard, A., and Sebban, M. Theoretical analysis of domain adaptation with optimal transport. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 737–753. Springer, 2017.

Schulman, J. and Lab, T. M. Lora without regret. *Thinking Machines Lab: Connectionism*, 2025. doi: 10.64434/tml.20250929. https://thinkingmachines.ai/blog/lora/.

S'ejourn'e, T., Peyr'e, G., and Vialard, F.-X. Unbalanced optimal transport, from theory to numerics. *ArXiv*, abs/2211.08775, 2022. URL https:

//api.semanticscholar.org/CorpusID:253553361.

Sinkhorn, R. A relationship between arbitrary positive matrices and doubly stochastic matrices. *Annals of Mathematical Statistics*, 35:876–879, 1964. URL https://api.semanticscholar.org/CorpusID:120846714.

Solomon, J., Greenewald, K., and Nagaraja, H. $k$-variance: A clustered notion of variance. *SIAM Journal on Mathematics of Data Science*, 4(3):957–978, 2022. doi: 10.1137/20M1385895.

Suzgun, M., Scales, N., Schärli, N., Gehrmann, S., Tay, Y., Chung, H. W., Chowdhery, A., Le, Q., Chi, E., Zhou, D., and Wei, J. Challenging BIG-bench tasks and whether chain-of-thought can solve them. In Rogers, A., Boyd-Graber, J., and Okazaki, N. (eds.), *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 13003–13051, Toronto, Canada, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.824. URL https://aclanthology.org/2023.findings-acl.824.

Thulke, D., Gao, Y., Pelser, P., Brune, R., Jalota, R., Fok, F., Ramos, M., Van Wyk, I., Nasir, A., Goldstein, H., et al. Climategpt: Towards ai synthesizing interdisciplinary research on climate change. *ArXiv preprint*, abs/2401.09646, 2024. URL https://arxiv.org/abs/2401.09646.

Villani, C. Optimal transport: Old and new. 2008. URL https://api.semanticscholar.org/CorpusID:118347220.

Wang, J., Lin, X., Qiao, R., Koh, P. W., Foo, C.-S., and Low, B. K. H. Nice data selection for instruction tuning in LLMs with non-differentiable evaluation metric. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=2wt8m5HUBs.

Wang, Y., Ma, X., Zhang, G., Ni, Y., Chandra, A., Guo, S., Ren, W., Arulraj, A., He, X., Jiang, Z., Li, T., Ku, M., Wang, K., Zhuang, A., Fan, R., Yue, X., and Chen, W. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. In Globersons, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J. M., and Zhang, C. (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/ad236edc564f3e3156e1b2feafb99a24-Abstract-Datasets_and_Benchmarks_Track.html.

Wei, K., Iyer, R., and Bilmes, J. Submodularity in data subset selection and active learning. In *International conference on machine learning*, pp. 1954–1963. PMLR, 2015.

Xia, M., Malladi, S., Gururangan, S., Arora, S., and Chen, D. LESS: selecting influential data for targeted instruction tuning. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=PG5fV50maR.

Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Ye, J., Liu, P., Sun, T., Zhan, J., Zhou, Y., and Qiu, X. Data mixing laws: Optimizing data mixtures by predicting language modeling performance. *ArXiv preprint*, abs/2403.16952, 2024. URL https://arxiv.org/abs/2403.16952.

Yin, J. O. and Rush, A. M. Compute-constrained data selection. *ArXiv preprint*, abs/2410.16208, 2024. URL https://arxiv.org/abs/2410.16208.

Zhao, H., Andriushchenko, M., Croce, F., and Flammarion, N. Long is more for alignment: A simple but tough-to-beat baseline for instruction fine-tuning. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024a. URL https://openreview.net/forum?id=0AZAjkXhit.

Zhao, T., Wang, S., Ouyang, C., Chen, M., Liu, C., Zhang, J., Yu, L., Wang, F., Xie, Y., Li, J., et al. Artificial intelligence for geoscience: Progress, challenges, and perspectives. *The Innovation*, 5(5), 2024b.

| Existing Work | Data Representation | Similarity | Selection Algorithm | Candidate Pool | Budget |
|---|---|---|---|---|---|
| Xia et al. (2024) | LESS | Cosine sim. | Doubly Greedy (DG) | Tulu V1 (Subset) | 13533 |
| Liu et al. (2024b) | LESS | L2 dist. | KNN-KDE, KNN-Unif. | Tulu V1 (Subset) | 13533 |
| Ivison et al. (2025) | RDS+ | Cosine sim. | Round-Robin (RR) | Tulu V2 | 10000 |

*Table 1.* **Overview of the fragmented literature on targeted instruction selection.** We summarize prior work and highlight differences in data representation, similarity metric, selection algorithm, candidate pool, and selection budget, which prevents us from systematically understanding which of these key factors contributes to performance.

## A. Fragmented Literature on Targeted Instruction Selection

Table 1 gives an overview of the fragmented literature on target instruction selection. We show that existing work, notably Xia et al. (2024), Liu et al. (2024b), and Ivison et al. (2025), widely vary in their data representations, similarity metrics, selection algorithm, candidate pool, and budget. While Xia et al. (2024) and Liu et al. (2024b) are similar in many ways, we observed that the selection algorithm in Liu et al. (2024b) makes additional changes to the data representation construction process and uses L2 distances between query and the candidates (Appendix K). These small but key differences prevent us from systematically comparing the selection algorithms, KNN-KDE and KNN-Uniform, introduced in Liu et al. (2024b). Finally, Ivison et al. (2025) uses Tulu V2 mixture (Ivison et al., 2023) as the candidate pool but also changes the data representation, similarity metric, and budget. For these reasons, we aim to bring clarity to this important but rather fragmented literature through systematic experiments.

## B. Target Tasks

We closely follow the target tasks from Ivison et al. (2025). In addition, we include the MMLU-Pro dataset (Wang et al., 2024). Table 2 provides statistics of the target tasks used in this work. Below, we provide descriptions of the target tasks:

**BBH (BIG-Bench Hard).** BBH is a curated subset of tasks from BIG-Bench that are empirically difficult for language models without advanced reasoning strategies. The benchmark includes tasks such as logical deduction, causal reasoning, temporal reasoning, symbolic manipulation, and algorithmic problem solving. BBH is commonly used to assess emergent reasoning abilities and sensitivity to prompting strategies, particularly few-shot and chain-of-thought prompting. We use the few-shot examples across all subtasks ($3 \times 27 = 81$ samples) as our query set. In the query set, all the few-shot examples are treated as individual examples. We follow the evaluation in (Suzgun et al., 2023) and evaluate the model with a 3-shot chain-of-thought. We report the average exact match on the test set.

**Codex.** Codex is an evaluation suite for coding-based models based on the correctness of completing coding tasks at scale, testing advanced reasoning ability over short and long-context coding tasks (Chen et al., 2021). We closely follow the custom query-set/test-set split from Ivison et al. (2025), which divides the existing dataset of 164 examples into a query set of 16 samples and uses the remaining examples as the test set. For downstream evaluation on the test set, we report the pass@10 with a sampling temperature of 0.8.

**GSM8K.** GSM8Kis a dataset of grade-school-level math word problems that require multi-step arithmetic reasoning. Each problem is paired with a detailed, step-by-step solution that explicitly outlines the reasoning process leading to the final answer. The dataset is widely used to evaluate numerical reasoning, chain-of-thought capabilities, and the ability of language models to perform symbolic manipulation and logical decomposition rather than surface-level pattern matching. We treat the 8-shot examples as individual samples and use them as our query set. We evaluate the test set with the 8 in-context samples using the chain-of-thought and report the exact match.

**TyDiQA.** TyDiQA is a multilingual question answering dataset built from Wikipedia that contains real information-seeking questions written by native speakers in 11 typologically diverse languages, avoiding translation artifacts. The dataset is widely used to evaluate multilingual and cross-lingual QA robustness, especially under linguistic diversity and distribution shift. We follow Ivison et al. (2025) and evaluate the models in a 1-shot setting across 9 languages where the answer is provided in the passage. We use the 1-shot samples across the languages as the query set.

| Target Task | Num. Query Set Samples | Num Test Set Samples | Target Task Evaluation |
|---|---|---|---|
| BBH (BIG-Bench Hard) | 81 | 6511 | 3-Shot |
| Codex | 16 | 148 | 0-Shot |
| GSM8K | 8 | 1319 | 8-Shot |
| TyDiQA | 9 | 5077 | 1-Shot |
| MMLU Pro | 70 | 12032 | 0-Shot |

*Table 2.* **Statistics of target tasks used in our experiments.** The query set is used to select samples from the candidate pool to train the base model, and then evaluated on the test set from the target task. Each target task is evaluated using 0 to N few-shot in-context examples, and we reuse the query set samples as these few-shot examples if mentioned in Appendix B.

**MMLU Pro.** MMLU-Pro is designed to evaluate advanced reasoning across a broad set of academic and professional domains, emphasizing multi-step reasoning and conceptual understanding, using carefully curated questions with reduced surface cues and strong controls against memorization or data leakage. It is commonly used to probe the upper limits of model reasoning performance in settings that more closely reflect expert-level problem solving. We treat the few-shot samples in the validation set as individual samples and use them as our query set. We perform zero-shot chain-of-thought evaluation on the test set and report exact match scores.

In contrast to Ivison et al. (2025), we do not include MMLU and SQuAD in the main experiments, as these datasets are largely saturated, making it harder to see clear performance trends. We also exclude AlpacaEval from the comparison because it relies on the GPT API as an evaluator, and conducting a large-scale study would be prohibitively expensive.

## C. Background on Optimal Transport

Optimal Transport (OT) is a principled approach for comparing probability distributions based on their underlying geometry, with strong theoretical guarantees (Villani, 2008). In machine learning, OT has been applied to a variety of domains, including domain adaptation (Courty et al., 2014), generative modeling (Arjovsky et al., 2017), and distance metrics (Alvarez-Melis & Fusi, 2020). The OT problem considers a complete metric space $\mathcal{X}$ with probability measures $\mu, \nu \in \mathcal{P}(\mathcal{X})$, which can be either discrete or continuous. The Kantorovich formulation of the transportation problem is defined as:

$$\text{OT}(\mu, \nu) = \min_{\pi \in \Pi(\mu,\nu)} \int_{\mathcal{X}\times\mathcal{X}} c(x, y) d\pi(x, y)$$

where $c(x, y)$ is a cost function over $\mathbb{R}^+$ and the set of couplings $\Pi(\mu, \nu)$ is defined as the joint probability distributions over the product space $\mathcal{X} \times \mathcal{X}$ with marginals $\mu$ and $\nu$ such that $\Pi(\mu, \nu) = \{\pi \in \mathcal{P}(\mathcal{X} \times \mathcal{X}) | P_{1\#}\pi = \mu, P_{2\#\pi} = \nu\}$. When $\mathcal{X}$ has a given distance metric $d_{\mathcal{X}}$, this is often treated as the cost function such that $c(x, y) = d_{\mathcal{X}}(x, y)^p$ for some $p \geq 1$. This is commonly defined as the $p-$Wasserstein distance where $W_p(\mu, \nu) = \text{OT}(\mu, \nu)^{\frac{1}{p}}$.

In the discrete optimal transport setting with finite samples, as considered in this work, let $\mu$ and $\nu$ be discrete probability measures defined as $\mu = \sum_{i=1}^{n} u_i \delta_{x_i}$ and $\nu = \sum_{j=1}^{m} v_j \delta_{y_j}$, where $\{x_i\}_{i=1}^{n} \in \mathcal{X}^n$ and $\{y_j\}_{j=1}^{m} \in \mathcal{Y}^m$ are points in a metric space (e.g., feature embeddings), with $u \in \mathbb{R}_+^n$, $v \in \mathbb{R}_+^m$, and $\sum_i u_i = \sum_j v_j = 1$. This is now a linear program and can be solved using classical solvers; however, its worst-case complexity is $O(N^3)$, which is often prohibitive in large-scale settings.

### C.1. Entropy-Regularized Optimal Transport

To improve computational efficiency, entropy regularization is commonly applied:

$$\text{OT}(\mu, \nu) = \min_{\pi \in \Pi(\mu,\nu)} \int_{\mathcal{X}\times\mathcal{X}} c(x, y) d\pi(x, y) + \varepsilon \text{H}(\pi | \mu \otimes \nu)$$

where $\text{H}(\pi|\mu \otimes \nu) = \int log(d\pi/d\mu d\nu)$ denotes the entropy of the transport plan. This objective is strongly convex and differentiable, and can be solved efficiently using the Sinkhorn–Knopp algorithm for the discrete case (Sinkhorn, 1964).

### C.2. Unbalanced Optimal Transport

Classical optimal transport assumes that the two measures have equal total mass. In many applications, such as noisy data, out-of-distribution comparison, or dataset selection, this assumption is undesirable. Unbalanced optimal transport relaxes the hard marginal constraints by penalizing deviations from the prescribed marginals (Benamou, 2003; Chizat et al., 2016; Liero et al., 2018; Peyré & Cuturi, 2018; S'ejourn'e et al., 2022).

The unbalanced optimal transport problem is defined as

$$\min_{\pi \in \mathcal{M}_+(\mathcal{X}\times\mathcal{X})} \int_{\mathcal{X}\times\mathcal{X}} c(x,y)\, d\pi(x,y) + \lambda_1\, D(P_{1\#}\pi \,\|\, \mu) + \lambda_2\, D(P_{2\#}\pi \,\|\, \nu),$$

where $P_{1\#}\pi$ and $P_{2\#}\pi$ denote the first and second marginals of $\pi$, and $D(\cdot\|\cdot)$ is a divergence between non-negative measures, commonly chosen as the Kullback–Leibler divergence.

Adding entropy regularization yields the unbalanced entropy-regularized OT objective:

$$\min_{\pi \in \mathcal{M}_+(\mathcal{X}\times\mathcal{X})} \int_{\mathcal{X}\times\mathcal{X}} c(x,y)\, d\pi(x,y) - \varepsilon H(\pi) + \tau_1\, \mathrm{KL}(P_{1\#}\pi \,\|\, \mu) + \tau_2\, \mathrm{KL}(P_{2\#}\pi \,\|\, \nu),$$

where $H(\pi) = -\int \log\!\left(\frac{d\pi}{dx\, dy}\right) d\pi$ denotes the entropy of the transport plan. As $\tau_1, \tau_2 \to \infty$, the marginal constraints are enforced exactly, recovering the balanced entropy-regularized OT formulation.

## D. Details on LESS

We now describe how LESS computes an influence matrix, which we use as a similarity matrix for our selection algorithms (see Appendix E.3 for implementation details). First, we perform LoRA warmup training of the model $f_{\theta_W}$ on a small randomly sampled warmup set $W \subseteq \mathcal{D}$ for $T$ epochs, saving the model parameters $\{\theta_W^{(t)}\}_{t=1}^{T}$ along with the corresponding optimizer states at the end of each epoch. Because LLMs are typically trained with Adam, LESS estimates influence via a first-order approximation of Adam training dynamics (Kingma & Ba, 2015), extending earlier first-order influence approximations developed for SGD (Pruthi et al., 2020). Concretely, LESS represents each example using low-dimensional projected features derived from (i) the query gradient and (ii) the candidate Adam update, where the Adam update is a preconditioned gradient computed from Adam's first- and second-moment estimates stored in the optimizer state. To make this scalable, we apply a random projection to these high-dimensional vectors to obtain compact representations in a lower-dimensional subspace. Finally, we compute influence for every query-candidate pair by aggregating cosine similarities between the projected query gradients and the projected candidate Adam updates across checkpoints, weighted by the average learning rate between checkpoints.

More formally, for each checkpoint $t$, we first compute (i) the query gradient and (ii) the candidate Adam update vector:

$$\nabla_\theta \ell\left(\theta_W^{(t)};\, q_i\right) \in \mathbb{R}^P,$$

and

$$\begin{aligned}
\Gamma\left(\theta_W^{(t)};\, z_j\right) &= \frac{\hat{m}^{\,t+1}}{\sqrt{\hat{v}^{\,t+1}} + \epsilon}, \\
g_t &:= \nabla_\theta \ell\left(\theta_W^{(t)};\, z_j\right), \\
\hat{m}^{\,t+1} &= \frac{\beta_1 m^t + (1-\beta_1)\, g_t}{1-\beta_1^{t+1}}, \\
\hat{v}^{\,t+1} &= \frac{\beta_2 v^t + (1-\beta_2)\, g_t^{\odot 2}}{1-\beta_2^{t+1}}.
\end{aligned}$$

Next, we apply a random projection $\Pi \in \mathbb{R}^{P\times d}$ to obtain low-dimensional representations:

$$\tilde{\nabla}\ell\left(\theta_W^{(t)};\, q_i\right) = \Pi^{\intercal} \nabla_\theta \ell\left(\theta_W^{(t)};\, q_i\right), \qquad \tilde{\Gamma}\left(\theta_W^{(t)};\, z_j\right) = \Pi^{\top} \Gamma\left(\theta_W^{(t)};\, z_j\right).$$

Then the influence of candidate sample $z_j$ on query sample $q_i$ is computed as the weighted cosine similarity aggregated across checkpoints:

$$\mathrm{Inf}_{\mathrm{Adam}}(q_i, z_j) \;=\; \sum_{t=1}^{T} \bar{\eta}_t \;\cos\Big(\tilde{\nabla}\ell\Big(\theta_W^{(t)};\, q_i\Big)\,, \tilde{\Gamma}\Big(\theta_W^{(t)};\, z_j\Big)\Big). \tag{2}$$

Here, $q_i$ is the $i$-th query sample, $z_j$ is the $j$-th candidate sample, $m^{(t)}$ and $v^{(t)}$ are the first and second moments from the saved checkpoint optimizer states obtained during warmup training, $\beta_1$, $\beta_2$, and $\epsilon$ are Adam-specific hyperparameters, and $\Pi$ is a random projection matrix whose entries are drawn from a Rademacher distribution (i.e., $\Pi_{ij} \sim \mathcal{U}(\{-1, 1\})$) (Johnson & Lindenstrauss, 1984; Park et al., 2023). $\bar{\eta}_t$ is the average learning rate between the $(t-1)$-th and $t$-th checkpoint. We compute $\mathrm{Inf}_{\mathrm{Adam}}(q_i, z_j)$ for all query-candidate pairs to obtain an $\mathbb{R}^{M\times N}$ influence matrix, which we use as the similarity matrix for instruction selection.

# E. Implementation Details: Data Representation

## E.1. RDS+

We use the weighted mean for RDS+ from Ivison et al. (2025) and Muennighoff (2022). For an input sequence $z$ of length $L$, we compute its representation as $z = \sum_{i=1}^{L} w_i h_i$, where $i$ is the token index, $h_i$ is the $i$-th hidden state from the base pre-trained language model, and $w_i = \frac{i}{\sum_{j=1}^{L} j}$ is the positional weight. We truncate both queries and candidates to a maximum length of 2048 tokens.

## E.2. EMBED

Following Ivison et al. (2025), we use GTR-T5 Base (Ni et al., 2022) to get data representations for the candidate pool and query set. For the examples in the query set, we also include an additional prefix `Instruct: Given a sample, find the passages closest to that sample. \nQuery: {query}`.

## E.3. LESS

We closely follow the LESS implementation of Xia et al. (2024). For warmup training, we sample 10,000 instruction–response pairs from the candidate pool and train the base model with LoRA (Hu et al., 2022) using cross-entropy loss. Following the original LESS setup for Llama 2 7B, we apply trainable LoRA parameters to all attention blocks during warmup. We follow the same LoRA setting for Llama 3.2 3B. For the remaining models, following recent recommendations for LoRA training (Schulman & Lab, 2025), we apply trainable LoRA parameters to both the attention and MLP blocks. We use the LoRA hyperparameters from Xia et al. (2024) (rank=128, $\alpha = 512$, dropout=0.1), and all other hyperparameters are provided in Appendix G. We train for 4 epochs and save each checkpoint (total of 316 steps).

Next, we compute gradients of the average loss over response tokens with respect to each LoRA checkpoint to produce the SGD update vectors for the query set and Adam update vectors for the candidate pool. We then apply a random projection using the TRAK package (Park et al., 2023) to map these update vectors into 8192-dimensional feature vectors, resulting in four sets of data representations for both the candidate pool and the query set. For fair comparison with RDS+ and EMBED, we do not average representations when query samples come from the same subtask; instead, we treat them as individual samples. We also compute the average learning rate across training steps between each checkpoint, normalize these values by dividing each by their sum, and finally compute the weighted cosine similarity between candidate and query representations to measure candidate influence on each query sample (Equation 2).

# F. Implementation Details: Selection Algorithms

## F.1. KNN KDE and KNN Uniform

We use the KNN-KDE implementation from TSDS (Liu et al., 2024b) in the released code. We reimplement KNN-Uniform from Algorithm 1 in their paper. We further simplify the implementation by using native PyTorch rather than the FAISS library to compute the distances between data representations. For a fair comparison with other methods, we use cosine distance rather than L2 distance between the query and candidate data representations. In KNN-KDE with LESS, we compute the cosine distance between candidate examples by modifying Equation 2 as $\mathbf{C_{ij}} = 1 - \sum_{t=1}^{T} \bar{\eta}_t \cos(\tilde{\nabla}\Gamma(z_i, f_{\hat{\boldsymbol{\theta}}_t}), \tilde{\nabla}\Gamma(z_j, f_{\hat{\boldsymbol{\theta}}_t}))$.

We closely followed the hyperparameters from Liu et al. (2024b) for KNN-KDE and KNN-Uniform (if the hyperparameter is used) as set $L = 5000$ (prefetching nearest neighbors), $K_{KDE} = 1000$ (KDE neighborhood size), $\sigma = 0.75$ (kernel bandwidth), and $C = 5.0$. We set $\alpha = 0.01$ as we wanted at least 10,000 candidates to have a transported mass greater than zero.

### F.2. Unbalanced OT (UOT)

Algorithm 1 describes the UOT selection algorithm, a new selection algorithm that explicitly solves the unbalanced optimal transport problem to obtain a transport plan, then selects the candidate samples with the highest mass transported. We use the `sinkhorn_unbalanced` implementation from the POT package (Flamary et al., 2021; 2024) to solve the optimization problem. We set $\varepsilon = 0.01$, $\tau_1 = \infty$, and $\tau_2 = 0.0001$. In all our experiments, we compute the cosine distance between the query set and candidate pool and normalize between 0 and 1 by dividing by 2. We use this normalized cosine distance matrix as our cost matrix.

**Algorithm 1** Unbalanced OT Selection

**Input:** Candidate pool $\mathcal{D} = \{z_i = (x_i, y_i)\}_{i=1}^{N}$, distance matrix (or cost matrix) between the query set and the candidate pool $\boldsymbol{C} \in \mathbb{R}^{M \times N}$, entropy regularization term $\varepsilon$, marginal relaxation terms $\tau_1$ and $\tau_2$ and budget $B \leq |\mathcal{D}|$.
**Output:** Selected subset $\hat{S}$.
$\Pi \leftarrow \mathrm{UOT}(\boldsymbol{C}, \varepsilon, \tau_1, \tau_2)$ // solve UOT (Appendix C) and get the transport plan
$\mathbf{r} \leftarrow \Pi\, \mathbf{1}_N$ // sum columns (row-sums): $r_j = \sum_{i=1}^{N} \Pi_{j,i}$
$\pi \leftarrow \mathrm{argsort}(-\mathbf{r})$ // indices sorted by descending $m_i$
$\hat{S} \leftarrow \emptyset$
**for** $k = 1$ to $B$ **do**
  $\hat{S} \leftarrow \hat{S} \cup \{z_{\pi_k}\}$ // select the top-$B$ candidates
**end for**
**return** $\hat{S}$

## G. Training Details and Hyperparameters

We follow a standard supervised fine-tuning pipeline and train the base model with a cross entropy loss over the response tokens. During supervised fine-tuning, we set the maximum sequence length to 2048 tokens. For this reason, we remove instruction-response pairs if the response does not appear in 2048 tokens to avoid a zero loss. The preprocessed candidate dataset contains about 198K examples. We apply the chat template to both the candidate and the query set during instruction selection, training, and evaluation, except during zero-shot evaluation of the pre-trained base language model.

| Hyperparameters | Values |
|---|---|
| Learning rate | 2e-5 |
| Learning rate scheduler | linear |
| Number of epochs | 2 |
| Warmup ratio | 0.03 |
| Optimizer | AdamW |
| Adam betas | (0.9, 0.999) |
| Adam epsilon | 1e-8 |
| Weight decay | 0.0 |
| Max. gradient norm | 1.0 |
| Max. sequence length | 2048 |
| Effective batch size | 128 |
| Mixed precision | bf16 |

*Table 3.* Hyperparameters used to train the base models on the selected instructions.

Table 3 lists all the hyperparameters used to train the base models on the selected data. We closely follow the training setup from Ivison et al. (2025). We run all the training and evaluation experiments on a single NVIDIA H100 GPU with 80GB of

memory.

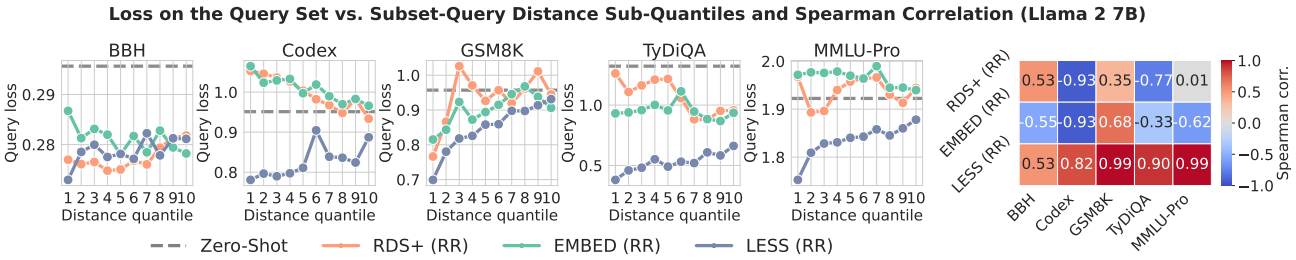


*Figure 9.* **Query loss across subset-query distance sub-quantiles and Spearman correlation.** We further stratify the first distance quantile from Section 5.1 into 10 sub-quantiles (1 = closest, 10 = farthest), select 500 examples per sub-quantile, and train the Llama 2 7B model. We report loss on the query set and Spearman correlation per dataset. LESS (RR) maintains a strong monotonic increase in loss with distance (high Spearman correlation), whereas RDS+ (RR) and EMBED (RR) show weak or inconsistent correlations.

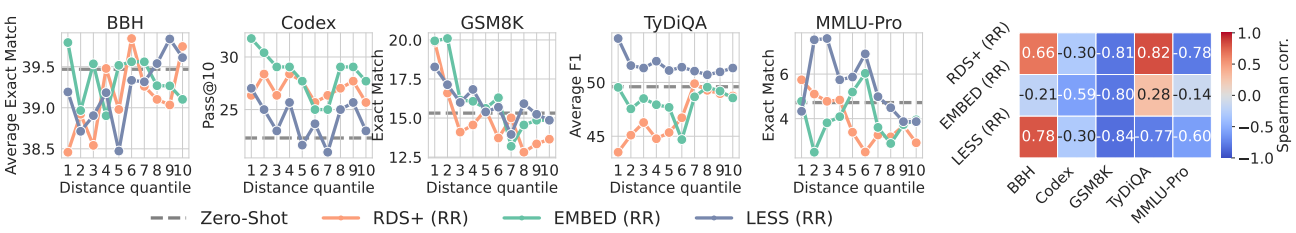


*Figure 10.* **Downstream performance across subset-query distance sub-quantiles and Spearman correlation.** Using the same sub-quantile construction and training protocol as Figure 9, we evaluate downstream task performance across sub-quantiles and report Spearman correlation per dataset (more negative is better). LESS (RR) shows a stronger negative correlation on average, but the performance differences are small, suggesting that many subsets within the closest quantile result in similar downstream performance.

## H. Fine-Grained Stratification of the Nearest Distance Quantile

Building on the distance quantile experiment (Section 5.1), we now aim to determine whether data representations can differentiate between very similar subsets, i.e., subsets that are closer to each other. Since we know the first distance quantile contains the samples most similar to the query set, we further subdivide it into 10 distance quantiles, select the top-K samples from each, and train the base models.

**Setup.** We closely follow the experiment setup from Section 5.1 for creating the distance quantiles, training, and evaluation. We further subdivide the closest distance quantile (first distance quantile) into 10 distance quantiles using the same procedure. Then, we select the top-500 samples from the distance quantiles and use them as training data to train Llama 2 7B.

**Results.** Figure 9 shows that LESS (RR) shows a high Spearman correlation across target tasks with the query loss. On the other hand, the RDS+ (RR) and EMBED (RR) do not show strong correlations across target tasks and often exhibit negative correlations with the distance sub-quantiles, suggesting that these data representations cannot differentiate between similar subsets. Figure 10 further shows that while LESS (RR), on average, shows a stronger negative Spearman correlation on the downstream evaluation compared to the other baselines, the difference in downstream performance appears to be similar. This suggests that multiple subsets in the first distance quantile can achieve a similar downstream performance.

## I. Differences between Selected Subsets

Here, we understand the differences between the selected subsets created using different data representations from Section 5.2. We compute average token length and Jaccard index across the selected subsets for query sets when the budget is 10,000 samples.

Figure 11 shows that subsets created with LESS (RR) contain shorter sequences than those produced by RDS+ (RR) and EMBED (RR) across all target tasks. We suspect this bias toward longer sequences may explain the competitive performance

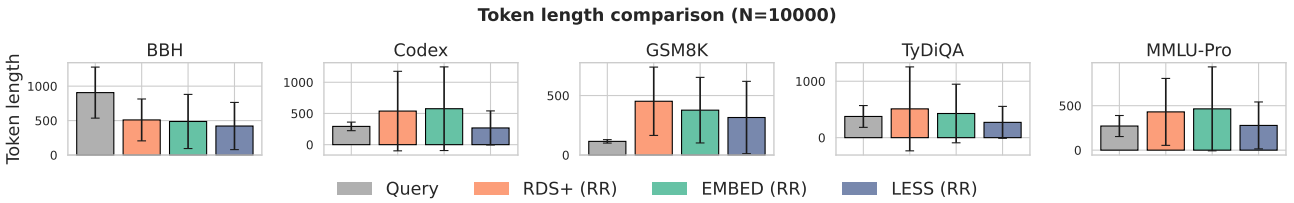


*Figure 11.* **Average token length of the query set and the selected subsets with different data representations.** We find that LESS (RR) is biased towards shorter sequences, whereas RDS+ (RR) and EMBED (RR) select subsets with longer sequences.

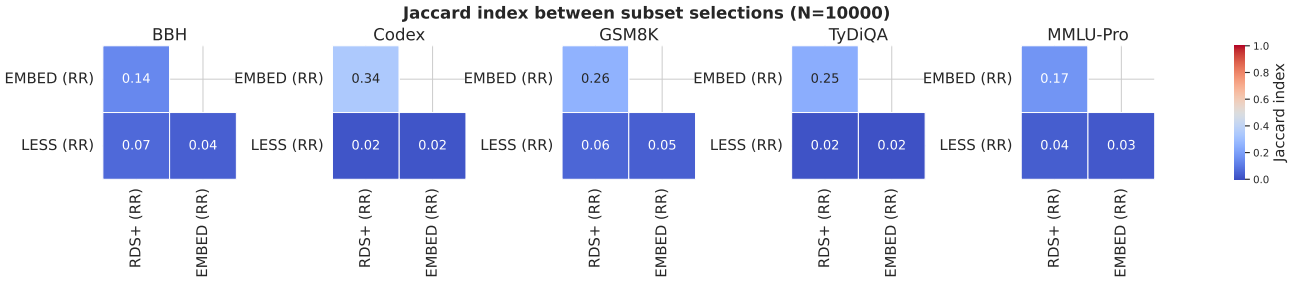


*Figure 12.* **Jaccard index between selected subsets created using different data representations.** We find that model-based representations (RDS+ and EMBED) have a higher Jaccard index compared to LESS.

of RDS+ (RR) and EMBED (RR) as we increase the budgets across target tasks (Zhao et al., 2024a). Finally, Figure 12 shows that RDS+ (RR) and EMBED (RR) have higher Jaccard indices compared to LESS (RR), suggesting that model-based embeddings share greater similarity in the examples they select.

## J. Cheaper Proxies for LESS

LESS data representations are computationally expensive as they require a forward and a backward pass over all the candidate samples. Here, we revisit the experiment of computing the LESS data representations from a smaller proxy model and then use them to select samples from the candidate pool to train the larger model (Appendix D.5 in Xia et al. (2024)). We broaden the set of proxy models, varying in size and include pre-trained LLMs trained with different token budgets to better understand how the performance of the larger base model relates to size and token budgets.

**Setup.** We consider the following proxy models: Pythia 160M (Biderman et al., 2023), SmolLM 135M (Allal et al., 2024), SmolLM2 135M (Allal et al., 2025), and Llama 3.2 3B (Grattafiori et al., 2024). We follow the same procedure to obtain the data representations for all proxy models (Appendix E.3 and Appendix G) and select samples for different budgets using a greedy round-robin approach. Then, we train Llama 2 7B on the selected instructions and report the downstream performance. We also include LESS without the proxy models, along with the Random baseline, to better contextualize the results.

**Results.** Figure 13 shows that, at higher budgets, smaller proxy models outperform the baseline LESS (RR, Llama 2 7B) on three out of the five target tasks. While these results suggest that proxy models for instruction selection are viable, similar to Xia et al. (2024), not all the proxy models perform well. We find that Pythia-160M performs as poorly as Random on several target tasks.

Next, we see that LESS (RR, SmolLM-135M) and LESS (RR, SmolLM2-135M) outperform or match Random across all target tasks, and sometimes even outperform LESS (RR, Llama 2 7B). A key difference between the two is that SmolLM-135M is pre-trained on 600B tokens, whereas SmolLM2-135M is pre-trained on 2 trillion tokens. We observe that on some target tasks, such as BBH and GSM8K, LESS (RR, SmolLM2-135M) achieves higher performance at low budgets, whereas on the rest, LESS (RR, SmolLM-135M) either matches or outperforms LESS (RR, SmolLM2-135M). These results show that the relationship between training tokens and proxy data representation for instruction selection remains unclear.

Finally, we observe that LESS (RR, Llama 3.2 3B) outperforms the baseline LESS (RR, Llama 2 7B) on three out of five

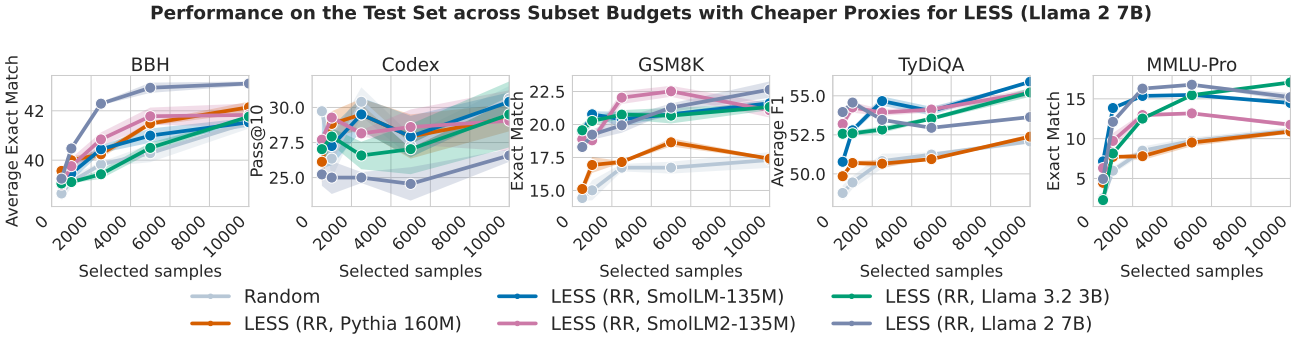


*Figure 13.* **Cheaper proxies for LESS (fixed selection algorithm).** With the same greedy round-robin selection procedure and the budgets from Section 5.2, we report downstream performance of Llama 2 7B when LESS representations are computed using proxy models, averaged across three seeds and the standard error. SmolLM-135M and SmolLM2-135M consistently match or outperform the Random baseline across target tasks, whereas Pythia-160M often matches Random on several tasks. Llama 3.2 3B matches or outperforms LESS computed with Llama 2 7B on multiple target tasks, suggesting a trade-off between proxy and target model size when approximating instruction selection.

target tasks, suggesting a trade-off between proxy and downstream model sizes in how well they approximate instruction selection of the larger models.

Overall, these results suggest that using proxy models for instruction selection to train larger models can dramatically reduce cost, but that a more thorough investigation is necessary to understand how to choose proxy models (Khaddaj et al., 2025).

## K. KNN-Uniform and KNN-KDE with L2 Distance

We now select instructions with KNN-Uniform and KNN-KDE selection algorithms and use L2 distances instead of cosine distance to match the original implementation in Liu et al. (2024b).

**Setup.** We use the Llama 2 7B LESS representations for both the query set and the candidate pool to compute distances. Instead of computing the weighted average between the query and the candidate representation (Equation 2), following Liu et al. (2024b), we scale the representation for the epoch checkpoint by the average learning rate for that epoch, concatenate all the scaled representations across epochs, and normalize them by the L2 norm. Then, we compute the L2 distance between the query and the candidate data representations. In KNN-KDE, we use the concatenated candidate data representations to compute L2 distances between them. We follow the same hyperparameters from Appendix K to select the instructions for a given budget. We report the downstream performance of these two selection methods across three seeds. We contextualize these results by comparing them with KNN-KDE and KNN-Uniform when implemented with cosine distance.

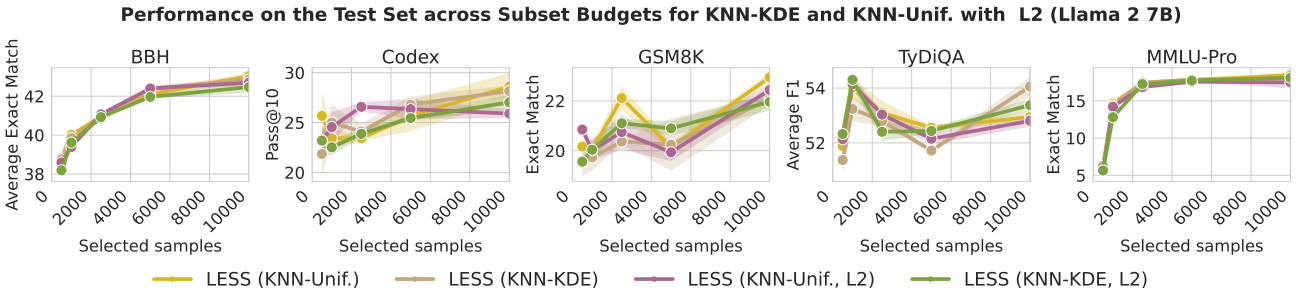


*Figure 14.* **KNN-Uniform and KNN-KDE with L2 distance (fixed data representation).** With the same Llama 2 7B LESS representations and the budgets from Section 5.3, we report downstream performance for KNN-Uniform and KNN-KDE when distances are computed with L2 (following Liu et al. (2024b)), averaged across three seeds and the standard error. We compare against cosine distance variants and find similar performance trends across budgets.

**Results.** Figure 14 shows that both the KNN-KDE and KNN-Uniform with L2 show similar performance trends when cosine distance is used to compute the distances, which further validates our conclusions regarding the choice of selection

algorithms in Section 5.3.

# L. Proofs

## L.1. Lemmas used for proofs of Theorem 6.1 and Theorem 6.2

**Lemma L.1** (**Domain adaptation bound**; adapted from Theorem 2 in (Redko et al., 2017))**.** *Let $S$ and $T$ be two samples of size $N_S$ and $N_T$ drawn i.i.d from probability measures $P_S$ and $P_T$ on $\mathbb{R}^d$, respectively. Let $\hat{P}_S := \frac{1}{N_S}\sum_{x_S \in S} \delta_{x_S}$ and $\hat{P}_T := \frac{1}{N_T}\sum_{x_T \in T} \delta_{x_T}$, where $\delta_x$ is the Dirac measure at $x$, be the associated empirical measures. Let $\ell$ be any loss function that is symmetric, convex, bounded, obeys the triangular inequality, and for $z = (x, y)$ has the parametric form $|y - f_\theta(x)|^q$ for some $q > 0$. Then, for any $d' > d$ and $c' < \sqrt{2}$, there exists some constant $N_0$ depending on $d'$ such that for any $\delta > 0$ and $\min(N_S, N_T) \geq N_0 \max(\delta^{-(d'+2)}, 1)$ with probability at least $1 - \delta$ for all datasets $D$ the following holds:*

$$L_T(\theta_D) \leq L_S(\theta_D) + W_1(\hat{P}_S, \hat{P}_T) + \sqrt{\frac{2}{c'}\log\left(\frac{1}{\delta}\right)}\left(\frac{1}{\sqrt{N_S}} + \frac{1}{\sqrt{N_T}}\right) + \lambda_{S,T} \tag{3}$$

where $\lambda_{S,T} := L_T(\theta) + L_S(\theta)$ for $\theta$ that minimizes the combined error $L_T(\theta) + L_S(\theta)$ over all possible model parameters $\theta$.

### Assumptions for Lemma L.2 and Theorem 6.2

A1. (**Strong convexity**) For any fixed dataset $D \subset \mathcal{Z}$, the empirical risk $L_D(\theta)$ is $\mu$–strongly convex in $\theta$.

A2. (**Smooth data dependence**) The gradient $\nabla_\theta \ell(z; \theta)$ is $G_{\theta z}$-Lipschitz with respect to the data point $z$, i.e., $\|\nabla_\theta \ell(\theta; z) - \nabla_\theta \ell(\theta; z')\| \leq G_{\theta z}\|z - z'\|$, for all $\theta$ in a relevant neighborhood.

A3. (**Lipschitzness at the ERM**) For a fixed dataset $D$, the empirical risk $L_D$ is $K$-Lipschitz with respect to the model parameter. That is, for any parameters $\theta$ and $\theta'$, $|L_D(\theta) - L_D(\theta')| \leq K\|\theta - \theta'\|$.

**Lemma L.2** (**Wasserstein stability bound**)**.** *Let $\mathcal{T}$ be a target dataset. Assume assumptions A1–A3 hold. Then, for any two datasets $D$ and $D'$, $|L_\mathcal{T}(\theta_D) - L_\mathcal{T}(\theta_{D'})| \leq C_\star\, W_1(\hat{P}_D, \hat{P}_{D'})$ with $C_\star := \frac{KG_{\theta z}}{\mu}$.*

*Proof.* Let $\theta_D = \arg\min_\theta L_D(\theta)$ and $\theta_{D'} = \arg\min_\theta L_{D'}(\theta)$. Using strong convexity once at $L_D$ (Assumption A1)

$$\mu||\theta_D - \theta_{D'}|| \leq ||\nabla L_D(\theta_D) - \nabla L_D(\theta_{D'})|| = ||\nabla L_D(\theta_{D'})||$$

where the last equality comes from $||\nabla L_D(\theta_D)|| = 0$ due to optimality. Similarly, because $||\nabla L_{D'}(\theta_{D'})|| = 0$ , then

$$\mu||\theta_D - \theta_{D'}|| \leq ||\nabla L_D(\theta_{D'}) - \nabla L_{D'}(\theta_{D'})||. \tag{4}$$

For a any fixed $\theta$, Assumption A2 and Kantorovich–Rubinstein duality (Arjovsky et al., 2017) gives

$$||\nabla_\theta L_D(\theta) - \nabla_\theta L_{D'}(\theta)|| \leq G_{\theta z} W_1(\hat{P}_D, \hat{P}_{D'}). \tag{5}$$

Combining (4) and (5)

$$||\theta_D - \theta_{D'}|| \leq \frac{G_{\theta z}}{\mu} W_1(\hat{P}_D, \hat{P}_{D'}). \tag{6}$$

Assumption A3 gives $|L_\mathcal{T}(\theta_D) - L_\mathcal{T}(\theta_{D'})| \leq K\|\theta_D - \theta_{D'}\|$, which combined with (6) gives

$$|L_\mathcal{T}(\theta_D) - L_\mathcal{T}(\theta_{D'})| \leq K||\theta_D - \theta_{D'}|| \leq \frac{K \cdot G_{\theta z}}{\mu} W_1(\hat{P}_D, \hat{P}_{D'}). \tag{7}$$

□

**Lemma L.3 (High-probability bound for the Wasserstein distance of an empirical measure).** *Let $\mathcal{D} = \{z_1, \ldots, z_N\} \subset \mathbb{R}^d$ lie in a set of diameter $\Delta$ and $d \geq 3$. Let $\mathcal{S}^{\text{rnd}} \subset \mathcal{D}$ be a uniform subset of size $B$ sampled without replacement, and let $\hat{P}_{\mathcal{D}}$ and $\hat{P}_{\mathcal{S}^{\text{rnd}}}$ be the empirical distributions on $\mathcal{D}$ and $\mathcal{S}^{\text{rnd}}$ respectively. Then there exists a constant $C_d > 0$ depending only on $d$ such that, for any $\delta \in (0, 1)$, with probability at least $1 - \delta$:*

$$W_1(\hat{P}_{\mathcal{S}^{\text{rnd}}}, \hat{P}_{\mathcal{D}}) \leq C_d \Delta B^{-1/d} + \Delta \sqrt{\frac{2\log(1/\delta)}{B}} \tag{8}$$

*Proof.* The proof proceeds by decomposing $W_1(\hat{P}_{\mathcal{S}^{\text{rnd}}}, \hat{P}_{D_c})$ into its expectation and its concentration around the mean. First, we use Theorem 1 in Fournier & Guillin (2015), which provides a bound on the expected Wasserstein convergence rate. For dimensions $d \geq 3$:

$$\mathbb{E}[W_1(\hat{P}_{\mathcal{S}^{\text{rnd}}}, \hat{P}_{\mathcal{D}})] \leq C_d \Delta B^{-1/d} \tag{9}$$

where $C_d$ is a constant depending only on dimension $d$.

Next, we view $f(\mathcal{S}^{\text{rnd}}) := W_1(\hat{P}_{\mathcal{S}^{\text{rnd}}}, \hat{P}_{\mathcal{D}})$ as a function of the random subset $\mathcal{S}^{\text{rnd}}$. Since the data lies in a bounded domain of diameter $\Delta$, changing a single data point in $\mathcal{S}^{\text{rnd}}$ changes the probability mass at that location by $1/B$, and moves it by a distance of at most $\Delta$. Therefore, the function $f$ satisfies the bounded difference property with constant $c_i = \Delta/B$. Then, by McDiarmid's inequality (McDiarmid, 1989), for any $\epsilon > 0$:

$$\mathbb{P}\left(W_1(\hat{P}_{\mathcal{S}^{\text{rnd}}}, \hat{P}_{\mathcal{D}}) - \mathbb{E}[W_1(\hat{P}_{\mathcal{S}^{\text{rnd}}}, \hat{P}_{\mathcal{D}})] \geq \epsilon\right) \leq \exp\left(\frac{-2\epsilon^2}{\sum_{i=1}^{B}(\Delta/B)^2}\right) = \exp\left(\frac{-2B\epsilon^2}{\Delta^2}\right) \tag{10}$$

Similar arguments have been used, for example, by Solomon et al. (2022), who apply McDiarmid's inequality to the Wasserstein distance viewed as a function of two random datasets, taking expectations over both. In contrast, in our setting, we fix $\mathcal{D}$ and treat the Wasserstein distance as a function of the random subset $\mathcal{S}^{\text{rnd}}$, taking the expectation over a single dataset. In this case, the bounded-differences condition required by McDiarmid's inequality follows directly from the stronger assumption that the support of $\mathcal{D}$, and hence the support of $\mathcal{S}^{\text{rnd}} \subseteq \mathcal{D}$, is bounded with diameter at most $\Delta$.

Setting the right hand side in (10) to $\delta$ and solving for $\epsilon$, we get $\epsilon = \Delta\sqrt{\frac{\ln(1/\delta)}{2B}}$. Thus, with probability at least $1 - \delta$:

$$W_1(\hat{P}_{\mathcal{S}^{\text{rnd}}}, \hat{P}_{\mathcal{D}}) \leq \mathbb{E}[W_1(\hat{P}_{\mathcal{S}^{\text{rnd}}}, \hat{P}_{\mathcal{D}})] + \Delta\sqrt{\frac{\ln(1/\delta)}{2B}} \tag{11}$$

Combining the expectation bound (9) with the concentration bound (11) yields the final result:

$$W_1(\hat{P}_{\mathcal{S}^{\text{rnd}}}, \hat{P}_{\mathcal{D}}) \leq C_d \Delta B^{-1/d} + \Delta\sqrt{\frac{2\log(1/\delta)}{B}} \tag{12}$$

□

### L.2. Theorem 6.1

Let $\ell : \Theta \times \mathcal{Z} \to \mathbb{R}_+$ be any loss function that is symmetric, convex, bounded, satisfies the triangle inequality, and for $z = (x, y)$ admits the parametric form $\ell(\theta; z) = |y - f_\theta(x)|^q$ for some $q > 0$. Let $\mathcal{D}$ denote a labeled candidate pool, and let $\mathcal{S} \subseteq \mathcal{D}$ be any subset of size $B := |\mathcal{S}|$. Let $\mathcal{Q}$ (the query set) and $\mathcal{T}$ (the test set) be labeled datasets. Then for any $c' < \sqrt{2}$, with probability at least $1 - 2\delta$:

$$L_{\mathcal{T}}(\theta_{\mathcal{S}}) \leq \underbrace{W_1(\hat{P}_{\mathcal{S}}, \hat{P}_{\mathcal{Q}})}_{\substack{\text{Subset and query} \\ \text{dataset distance}}} + \underbrace{W_1(\hat{P}_{\mathcal{Q}}, \hat{P}_{\mathcal{T}})}_{\substack{\text{Query and test} \\ \text{dataset distance}}} + \underbrace{L_{\mathcal{S}}(\theta_{\mathcal{S}})}_{\text{training error}} + \zeta\sqrt{\frac{2}{c'}\log\left(\frac{1}{\delta}\right)} + \tilde{\lambda} \tag{13}$$

where $W_1$ is the 1-Wasserstein distance with respect to the underlying metric on the embedding space $\mathcal{Z}$, $\zeta$ is a constant determined by the size of the datasets, with $\zeta = B^{-\frac{1}{2}} + 2|\mathcal{Q}|^{-\frac{1}{2}} + |\mathcal{T}|^{-\frac{1}{2}}$, and $\tilde{\lambda}$ is the combined error of the dataset $\tilde{\mathcal{S}} \subseteq \mathcal{D}$ of size $B$ that minimizes the combined error of $L_{\mathcal{S}}(\theta_{\tilde{\mathcal{S}}}) + 2L_{\mathcal{Q}}(\theta_{\tilde{\mathcal{S}}}) + L_{\mathcal{T}}(\theta_{\tilde{\mathcal{S}}})$.

*Proof.* The proof proceeds by applying Lemma L.1 twice. We first apply it to $L_{\mathcal{T}}(\theta_{\mathcal{S}})$, treating $\mathcal{T}$ as the target dataset and $\mathcal{Q}$ as the source dataset, with the function obtained by training on $\mathcal{S}$. We then apply it to $L_{\mathcal{Q}}(\theta_{\mathcal{S}})$, now treating $\mathcal{Q}$ as the target dataset and $\mathcal{S}$ as the source dataset, again using the prediction function trained on $\mathcal{S}$.

$$L_{\mathcal{T}}(\theta_{\mathcal{S}}) \leq L_{\mathcal{Q}}(\theta_{\mathcal{S}}) + W_1(\hat{P}_{\mathcal{Q}}, \hat{P}_{\mathcal{T}}) + \sqrt{\frac{2}{c'}\log\left(\frac{1}{\delta}\right)}\left(\frac{1}{\sqrt{|\mathcal{Q}|}} + \frac{1}{\sqrt{|\mathcal{T}|}}\right) + \lambda_{\mathcal{T},\mathcal{Q}} \tag{14}$$

$$\leq L_{\mathcal{S}}(\theta_{\mathcal{S}}) + W_1(\hat{P}_{\mathcal{S}}, \hat{P}_{\mathcal{Q}}) + W_1(\hat{P}_{\mathcal{Q}}, \hat{P}_{\mathcal{T}}) + \sqrt{\frac{2}{c'}\log\left(\frac{1}{\delta}\right)}\left(\frac{1}{\sqrt{B}} + \frac{2}{\sqrt{|\mathcal{Q}|}} + \frac{1}{\sqrt{|\mathcal{T}|}}\right) + \lambda_{\mathcal{T},\mathcal{Q}} + \lambda_{\mathcal{Q},\mathcal{S}} \tag{15}$$

$$= L_{\mathcal{S}}(\theta_{\mathcal{S}}) + W_1(\hat{P}_{\mathcal{S}}, \hat{P}_{\mathcal{Q}}) + W_1(\hat{P}_{\mathcal{Q}}, \hat{P}_{\mathcal{T}}) + \Gamma\sqrt{\frac{2}{c'}\log\left(\frac{1}{\delta}\right)} + \tilde{\lambda} \tag{16}$$

□

### L.3. Theorem 6.2

Let the candidate pool $\mathcal{D} \subset \mathbb{R}^d$ lie in a set of diameter $\Delta$ and $d \geq 3$. Let $\mathcal{S}^{\text{rnd}} \subseteq \mathcal{D}$ be a subset of size $B$ sampled uniformly at random, and let $S_W^* \subseteq \mathcal{D}$ be a subset of size $B$ that minimizes the 1-Wasserstein distance to the query set $\mathcal{Q}$, i.e., $\mathcal{S}_W^* \subset \arg\min_{\mathcal{S}\subseteq\mathcal{D};|\mathcal{S}|=B} W_1(\hat{P}_{\mathcal{S}}, \hat{P}_{\mathcal{Q}})$. Assume assumptions A1–A3 hold. Then, there exists a constant $C_d > 0$, depending only on the dimension $d$, such that, with probability at least $1 - 2\delta$:

$$L_{\mathcal{T}}(\theta_{\mathcal{S}^{\text{rnd}}}) - L_{\mathcal{T}}(\theta_{S_W^*}) \leq C_\star\Bigg(\underbrace{C_d\Delta B^{-1/d}}_{\substack{\text{Curse of}\\ \text{dimensionality}}} + \underbrace{\Delta\sqrt{\frac{2\log(1/\delta)}{B}}}_{\text{Concentration bound}} + \underbrace{W_1(\hat{P}_{\mathcal{D}}, \hat{P}_{\mathcal{Q}})}_{\text{Pool-query mismatch}} + \underbrace{W_1(\hat{P}_{S_W^*}, \hat{P}_{\mathcal{Q}})}_{\text{Distance residual}}\Bigg)$$

*Proof.*

$$L_{\mathcal{T}}(\theta_{\mathcal{S}^{\text{rnd}}}) - L_{\mathcal{T}}(\theta_{S_W^*}) \leq C_\star W_1(\hat{P}_{\mathcal{S}^{\text{rnd}}}, \hat{P}_{S_W^*}) \tag{17}$$

$$\leq C_\star\Big(W_1(\hat{P}_{\mathcal{S}^{\text{rnd}}}, \hat{P}_{\mathcal{Q}}) + W_1(P_{S_W^*}, \hat{P}_{\mathcal{Q}})\Big) \tag{18}$$

$$\leq C_\star\Big(W_1(\hat{P}_{\mathcal{S}^{\text{rnd}}}, \hat{P}_{\mathcal{D}}) + W_1(\hat{P}_{\mathcal{D}}, \hat{P}_{\mathcal{Q}}) + W_1(P_{S_W^*}, \hat{P}_{\mathcal{Q}})\Big) \tag{19}$$

$$\leq C_\star\Big(C_d\Delta B^{-1/d} + \Delta\sqrt{\frac{2\log(1/\delta)}{B}} + W_1(\hat{P}_{\mathcal{D}}, \hat{P}_{\mathcal{Q}}) + W_1(\hat{P}_{\mathcal{S}^{\text{rnd}}}, \hat{P}_{\mathcal{Q}})\Big) \tag{20}$$

□

Line 17 follows from applying Lemma L.2, line 18 applied triangle inequality by introducing the distance to the empirical distribution of $\mathcal{Q}$, line 19 again uses triangle inequality by introducing the distance to the empirical distribution of $\mathcal{D}$, and 20 is a result of applying Lemma L.3 on $W_1(\hat{P}_{\mathcal{S}^{\text{rnd}}}, \hat{P}_{\mathcal{D}})$.

## M. Ablations

We run ablations for experiments in Section 5 across four models of different sizes: Llama 3.2 3B (Grattafiori et al., 2024), SmolLM3 3B (Bakouch et al., 2025), Qwen3 4B Base (Yang et al., 2025), and Olmo 3 7B (Olmo et al., 2025).

Below, we summarize our key takeaways from the experiment across models (See Figures 15, 16, 17, and 18):

**Distance quantile experiment.** Only LESS (RR) creates subsets whose quantile distances strongly correlate with the loss. However, with over-trained models such as SmolLM3 3B and Qwen3 4B Base, the performance trends for LESS (RR) on datasets such as BBH and Codex show less correlation with the distance quantiles.

**Effect of data representation across subset budgets.** Although no single data representation with a fixed selection algorithm selects subsets that always perform the best, LESS with greedy round robin performs the best across all models on most of the target tasks compared to other model-based representation baselines. But on some datasets and models (BBH

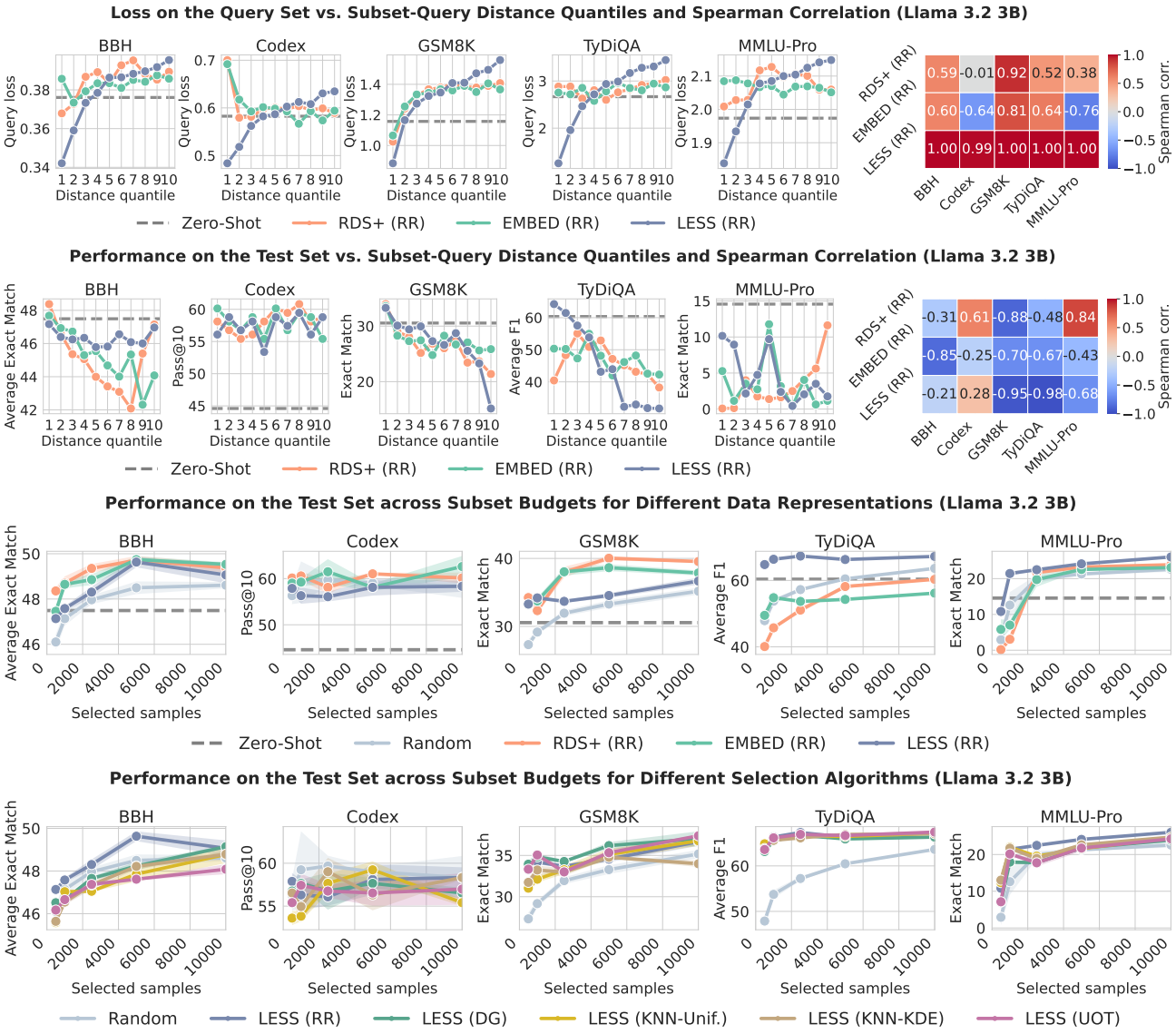


*Figure 15.* Ablation experiments with Llama 3.2 3B.

with SmolLM3, MMLU Pro with Qwen3 4B Base), we observe that the zero-shot baseline outperforms all instruction selection methods, suggesting that either the candidate pool does not have sufficient training examples to further improve the performance, or due to an accidental leakage of the target task in the pre-training corpus (Olmo et al., 2025).

**Effect of selection algorithm across subset budgets.** We observe that the performance trends for different selection algorithm generalizes to other models. For instance, LESS (RR) achieves the strongest results on BBH with Llama 3.2 3B and Qwen3 4B Base. We also find that, at higher budgets, LESS (UOT) delivers the best MMLU-Pro performance with SmolLM3 and Olmo3 7B Base. Overall, these findings indicate that, for certain target tasks, selection algorithms tend to exhibit consistent performance across different models.

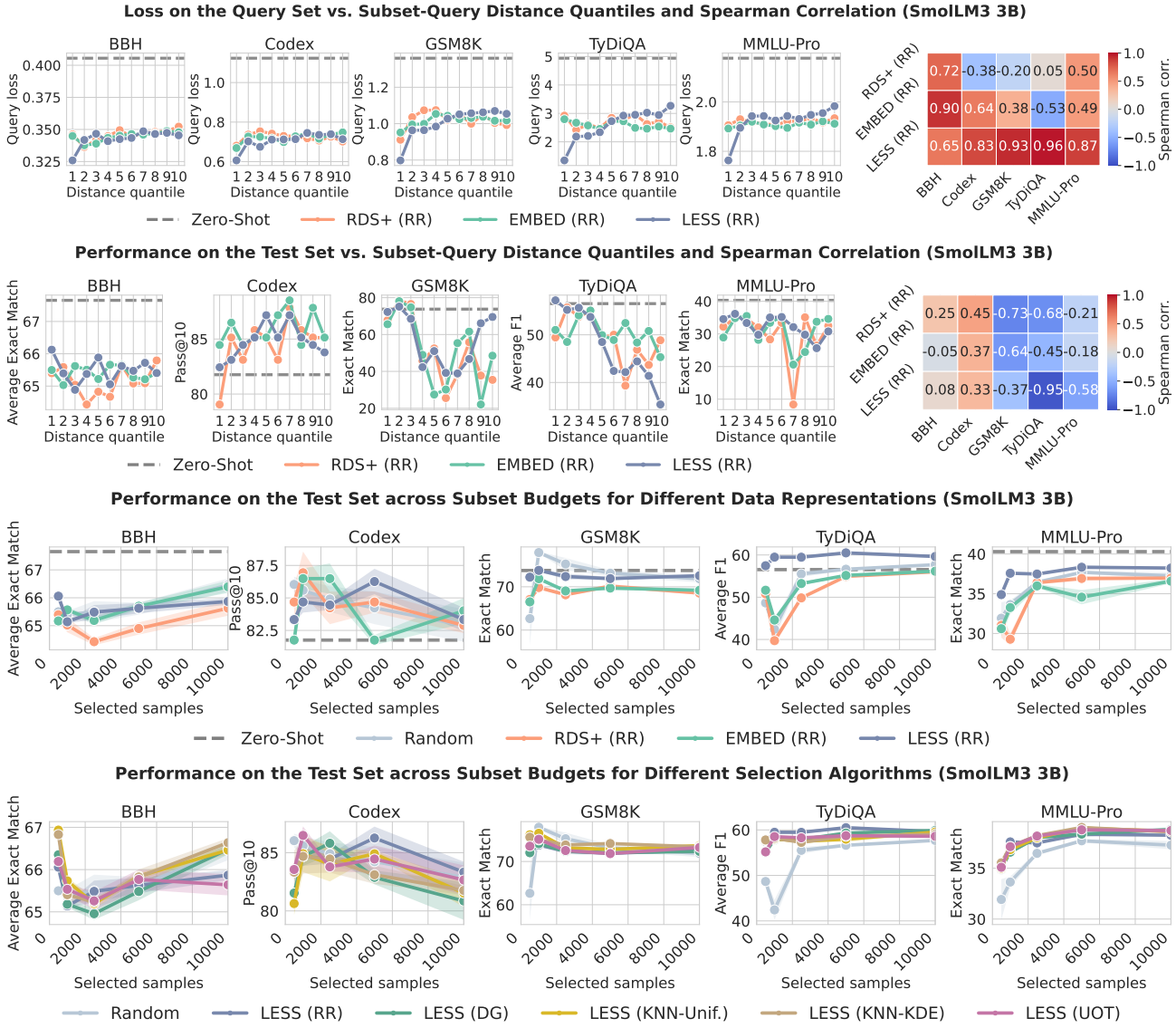


*Figure 16.* Ablation experiments with SmolLM3 3B.

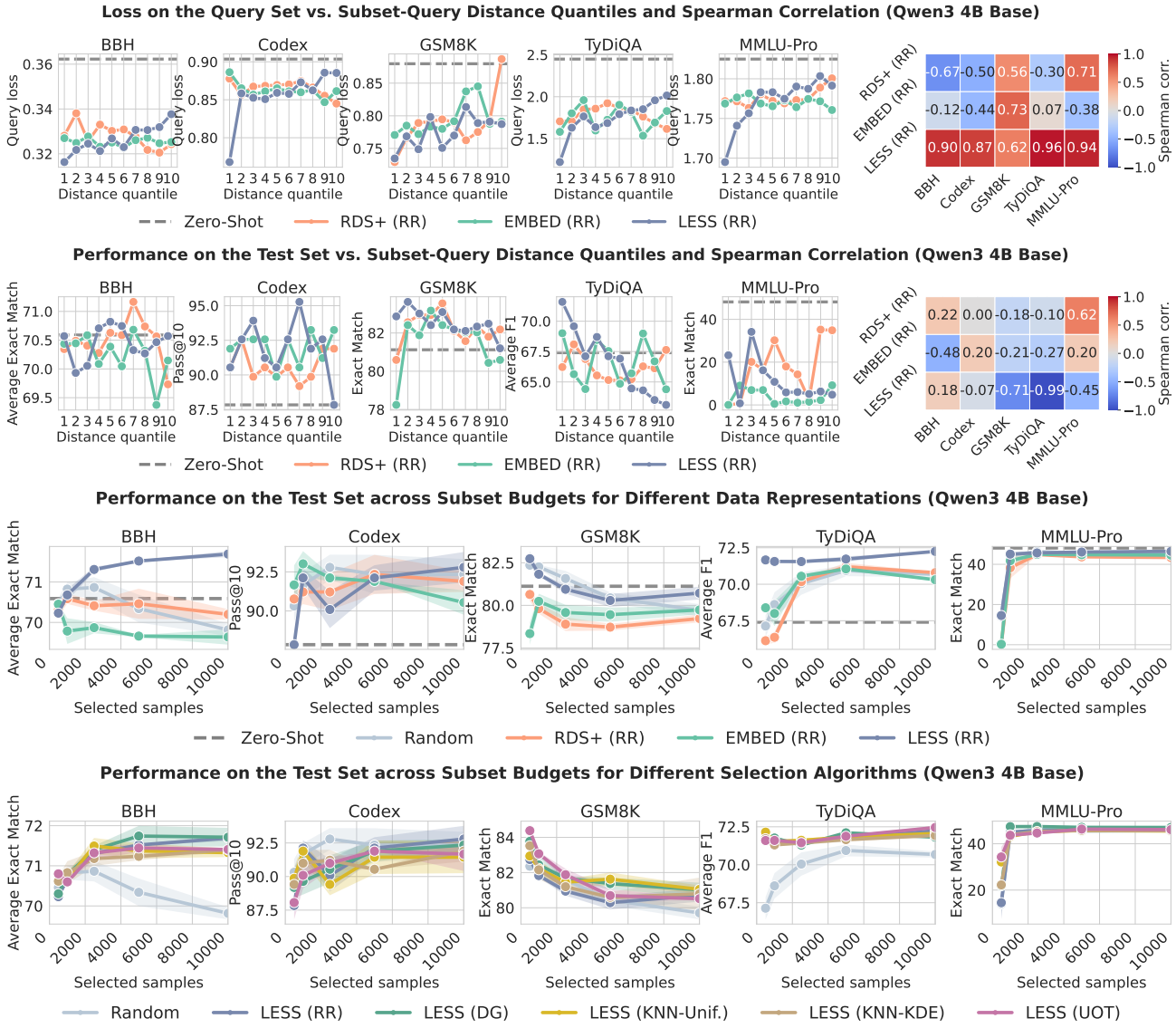


*Figure 17.* Ablation experiments with Qwen3 4B.

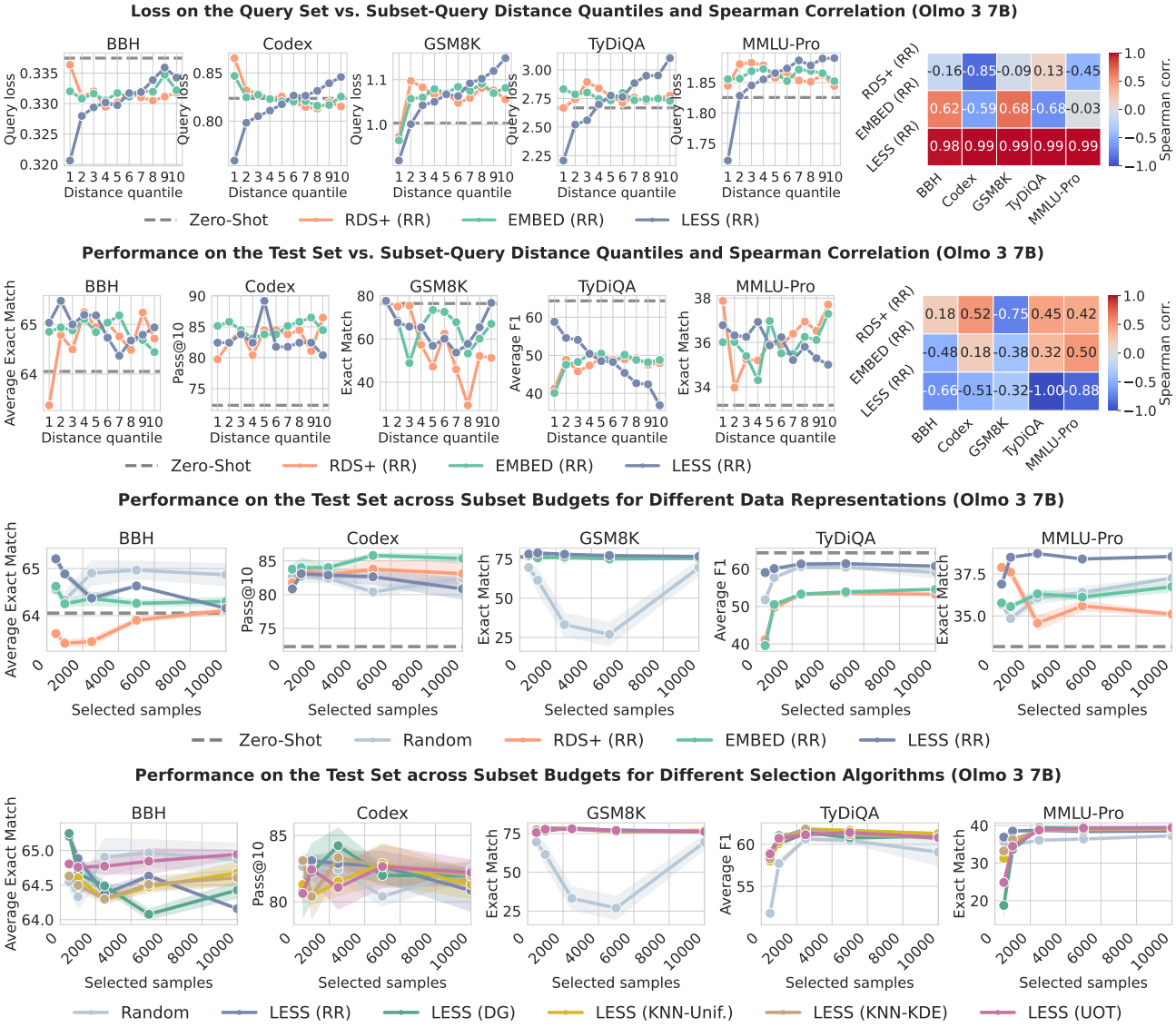


*Figure 18.* Ablation experiments with Olmo 3 7B.