Improving Task Diversity in Label Efficient Supervised Finetuning of LLMs

Abhinav Arabelly[*] **Jagrut Nemade**[*] **Robert D. Nowak** **Jifan Zhang**
University of Wisconsin–Madison
{arabelly, jnemade, rdnowak, jzhang769}@wisc.edu

Abstract

Large Language Models (LLMs) have demonstrated remarkable capabilities across diverse domains, but developing high-performing models for specialized applications often requires substantial human annotation — a process that is time-consuming, labor-intensive, and expensive. In this paper, we address the label-efficient learning problem for supervised finetuning (SFT) by leveraging task-diversity as a fundamental principle for effective data selection. This is markedly different from existing methods based on the prompt-diversity. Our approach is based on two key observations: 1) task labels for different prompts are often readily available; 2) pre-trained models have significantly varying levels of confidence across tasks. We combine these facts to devise a simple yet effective sampling strategy: we select examples across tasks using an inverse confidence weighting strategy. This produces models comparable to or better than those trained with more complex sampling procedures, while being significantly easier to implement and less computationally intensive. Notably, our experimental results demonstrate that this method can achieve better accuracy than training on the complete dataset (a 4% increase in MMLU score). Across various annotation budgets and two instruction finetuning datasets, our algorithm performs at or above the level of the best existing methods, while reducing annotation costs by up to 80%.

1 Introduction

Large Language Models have demonstrated remarkable capabilities across a diverse range of tasks and domains. However, for challenging tasks where LLMs still struggle, finetuning often requires a large number of human annotations and demonstrations to guide models toward correct behavior (Qin et al., 2024; Zhou et al., 2025; Chen et al., 2025). This annotation process is typically time-consuming, labor-intensive, and expensive, creating a significant barrier to developing high-performing models for specialized applications.

In this work, we address the label-efficient learning problem, where ground-truth responses are initially unknown. The goal is to develop methods that reduce annotation requirements while improving model performance. Our approach leverages task-level information as a fundamental organizing principle for effective data selection in instruction tuning.

The label-efficient SFT (Bhatt et al., 2024) and data selection literature (Yang et al., 2024; Wang et al., 2024) have introduced various diversity-based methods based on facility location, clustering, optimal design, and determinantal point process methods to ensure comprehensive data coverage. Despite their improved performances, these approaches face several practical challenges: (1) they typically rely on pre-computed embeddings that inadequately capture task-specific semantics; (2) their computational complexity scales poorly with dataset size, limiting applicability to large-scale problems; and (3) achieving optimal performance often requires careful hyperparameter tuning, adding complexity to implementation.

Our study introduces a novel approach that combines task-diversity principles with uncertainty quantification. Unlike existing methods that focus primarily on prompt-diversity in the embedding space, we leverage the inherent task categorizations (Kung et al., 2023; Ivison et al., 2022; Wang et al., 2022; Wei et al., 2022) widely available in modern LLM development datasets. Prompt here refers to individual input examples to an LLM. We define tasks according to their original data curation sources, such as various domains (e.g., medical knowledge, mathematics) and desired skills (e.g., summarization, translation). For instance, the FLAN dataset (Longpre et al., 2023)

[*] Equal contribution.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 21558–21570
November 4-9, 2025 ©2025 Association for Computational Linguistics

combines instruction tuning data from over a hundred sources across 1,691 tasks, while the Dolly dataset (Conover et al., 2023) covers eight distinct tasks, such as brainstorming, closed QA, information extraction and others.

Our approach is straightforward yet effective: we allocate a minimum amount of allocation budget across all tasks to ensure diversity, while allocating more budget to uncertain tasks, using an inverse confidence weighting strategy. The confidence is derived from the base model's confidence in answering the questions. This method achieves high diversity in data selection while prioritizing tasks where the model exhibits greater uncertainty, producing models that outperform those trained with existing sampling procedures, while being significantly more accessible and easier to implement.

Experimental results demonstrate that by prioritizing the labeling of a strategic subset of examples, our method achieves better accuracy than training on the complete dataset (a 4% increase in MMLU score). Across various annotation budgets and two instruction finetuning datasets, our algorithm consistently performs at or above the level of the best existing methods, while reducing annotation costs by up to 80%. These findings highlight the effectiveness of combining task diversity with uncertainty quantification as a guiding principle for data selection in instruction tuning and provide a practical approach for optimizing the annotation process without sacrificing model performance.

2 Related Work

2.1 LLM Supervised Finetuning

Supervised finetuning (SFT) has become a pivotal technique for aligning Large Language Models (LLMs) with human preferences and specific tasks (Ouyang et al., 2022; Wei et al., 2022; Touvron et al., 2023). This approach enhances pre-trained language models by training them on carefully curated prompt-response pairs, enabling more accurate instruction following and appropriate response generation. However, traditional SFT methods typically demand extensive human-annotated examples, creating resource-intensive processes that potentially limit scalability. This constraint has spurred research into more efficient finetuning approaches that can achieve comparable performance with significantly fewer labeled examples.

2.2 Label-Efficient Learning for LLMs

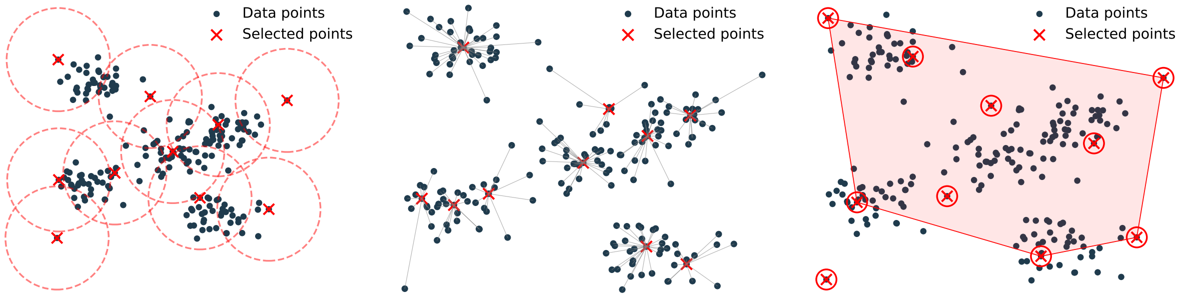
The pursuit of label-efficient learning for LLMs has garnered substantial attention as researchers strive to reduce annotation burdens while maintaining model performance. Various methodologies have emerged, including active learning and experimental design (i.e., one-batch active learning). In this paper, we focus on the one-batch active learning problem, similar to Bhatt et al. (2024), where the algorithm selects a single set of informative examples for annotation. This approach offers distinct advantages: it reduces logistical complexities compared to iterative active learning and provides computational benefits by eliminating expensive retraining cycles for LLMs.

Distinction from Data Selection. It is essential to distinguish label-efficient learning from data selection. While data selection operates with prior knowledge of ground truth responses, label-efficient learning functions without access to these annotations. Numerous data selection methods (Chen et al., 2023; Bukharin and Zhao, 2023; Du et al., 2023; Yang et al., 2024; Wang et al., 2024) incorporate ground truth responses as integral components of their selection procedures, rendering them unsuitable for scenarios where annotations have not yet been collected. Our approach addresses this limitation by focusing on task characteristics that can be evaluated prior to annotation.

Unifying Principles. Although data selection methods cannot be directly applied to label-efficient settings, certain algorithmic principles are shared between these problem domains. Existing methods in active learning aim to either enhance diversity among selected examples or reduce model uncertainty by prioritizing the most uncertain or incorrectly predicted examples (Lewis, 1995; Tong and Koller, 2001; Settles, 2009; Kremer et al., 2014; Ash et al., 2019, 2021; Citovsky et al., 2021; Mohamadi et al., 2022; Zhang et al., 2022; Nuggehalli et al., 2023; Zhang et al., 2023; Xie et al., 2024). Below, we provide a concise overview of key algorithms in this space for LLMs.

2.2.1 Diversity-Based Methods

Diversity-based data selection aims to construct representative subsets that ensure broad coverage of the input space by prioritizing diversity among the prompts selected for annotation. Below, we review three prominent diversity-based selection strategies: K-Center, Facility Location, and Determinantal Point Processes (DPPs). We use



(a) K-Center selection (b) Facility Location selection (c) DPP selection

Figure 1: Comparison of three diversity-based data selection baseline algorithms. In contrast to task diversity in Figure 2, these methods primarily rely on prompt diversity in the embedding space. All of these methods aim to cover the space of examples based on different criterions.

$Z = z_1, ..., z_m$ to denote the set of data examples in some embedding space, and the goal is to select the subset $S \subset Z$ to ensure good coverage of examples.

K-Center (Sener and Savarese, 2018) selects k data points to serve as centers of equal-radius balls in the representation space. The objective is to cover all data points using the smallest possible radius, ensuring that each example lies within the ball of at least one selected center (see Figure 1a). This approach encourages the selection of a subset that minimizes the maximum distance between any data point and its closest selected center. Formally, the objective function is:

$$S = \arg\min_{\substack{S' \subset X \\ |S'|=k}} \max_{z_i \in Z} \min_{z_j \in S'} \|z_i - z_j\|$$

As illustrated in Figure 1a, this coreset sampling method is vulnerable to outliers, as remote singular examples are often selected.

Facility Location (FL) (Mirchandani and Francis, 1990; Wei et al., 2015; Mirzasoleiman et al., 2020; Bilmes, 2022; Bukharin and Zhao, 2023; Bhatt et al., 2024) addresses the outlier issue by considering the average distance of examples to their corresponding centers rather than the worst case distance. This strategy utilizes a similarity kernel measure as a proxy for distance metrics, denoted by $\mathcal{K}(\cdot, \cdot)$, which quantifies the similarity between features z_i and z_j. The facility location objective is formulated as:

$$S = \arg\max_{\substack{S' \subset X \\ |S'|=k}} \sum_{z_i \in X} \max_{z_j \in S'} \mathcal{K}(z_i, z_j)$$

In this formulation, each data point $z_i \in X$ (considered a "client") is assigned to its most similar center $z_j \in S'$ (the "facility"), and the objective maximizes the cumulative similarity across all such assignments. The kernel function can be an l_2 distance as in k-center selection, or alternatively an RBF kernel with hyperparameter γ (Bhatt et al., 2024).

Determinantal Point Processes (DPP) provide a probabilistic framework for subset selection that maximizes data coverage (Kulesza et al., 2012; Wang et al., 2024). A DPP is inherently a stochastic process where the probability of selecting subset S is proportional to:

$$\det(K(S)) := \det([z]_{z \in S}^{\top} [z]_{z \in S})$$

where $K(S)_{i,j} = \langle z_i, z_j \rangle$ for the i-th and j-th elements in S. This determinant quantifies the volume of the parallelotope formed by the embeddings in S when the selection budget k is no larger than the feature dimensionality d. Beyond inner products, the kernel map can be replaced with other kernels $K(S)_{i,j} = \mathcal{K}(z_i, z_j)$ similar to FL. In Wang et al. (2024), the authors employ a deterministic version of DPP, equivalent to D-optimal design: $\arg\max_S \log\det(\mathcal{K}(S))$. As shown in Figure 1c, this method tends to select examples along the boundary of the prompt distribution. For data selection, Wang et al. (2024) also incorporates quality scores based on ground-truth answers. Since we lack access to such information in label-efficient SFT, we instead integrate confidence scores as proposed by Bhatt et al. (2024) throughout our experiments.

2.2.2 Uncertainty-Based Methods

Uncertainty-based selection strategies identify examples where the model exhibits low confidence, utilizing metrics such as entropy, token probability, and margin-based measures to quantify prediction uncertainty (Settles, 2009; Lewis, 1995). Recent research has explored more sophisticated approaches specifically tailored for large language models, advancing beyond simple uncertainty estimates. Bhatt et al. (2024) extended entropy-, confidence-, and margin-based scores to SFT of LLMs. Similar to us, Kung et al. (2023) also utilizes the task definitions from existing datasets, and developed methods that prioritize labeling tasks with the highest uncertainties. However, their algorithm would spend all of their budgets on the most difficulty tasks, thus reducing the data coverage and diversity in selected prompts. Our work draws inspiration from these techniques while introducing a simpler and more practical approach that focuses on task diversity and model confidence. This provides an effective method for optimizing annotation resources without requiring complex embeddings or algorithms.

3 Problem Formulation

We adopt the label-efficient SFT framework introduced by Bhatt et al. (2024). In this framework, the learner is provided with an initial set of N prompts $X = \{x_1, x_2, \ldots, x_N\}$, where each prompt x_i consists of l tokens, i.e., $x_i = \{x_{i,1}, \ldots, x_{i,l}\}$. Let g represent the pretrained language model. Given a limited annotation budget $k < N$, our goal is to develop effective selection strategies that identify a subset $S \subset X$ of the most informative prompts, such that $|S| = k$.

After selecting subset S, high-quality responses are collected for each prompt $x \in S$ from annotators (such as human experts or advanced LLMs). These annotated prompt-response pairs are then used to finetune model g. Let g' denote the resulting model after SFT on these annotated pairs. Our objective is to maximize the performance improvement of g' while minimizing the required annotation budget.

Throughout this paper, we use feature embeddings extracted from prompts using the base model. Let $f : X \to \mathbb{R}^d$ denote the feature mapping between prompts to embedding, all of the diversity based methods in Section 2.2.1 can be viewed as using $z_i := f(x_i)$. Concretely, we extract these feature embeddings from the penultimate layer's hidden state of the pretrained model g during inference on the prompt.

4 Methods

4.1 Task Definition

Instruction tuning datasets today are often consisted of carefully curated prompt/response pairs across numerous domain and tasks. In this paper we predominantly experiment with a 90K subset of the FLAN V2 data (Longpre et al., 2023) and the entire Dolly dataset (Conover et al., 2023) (with responses initially masked to simulate the label-efficient learning scenario).

The FLAN V2 dataset provides a broad and diverse collection of examples across 1,691 tasks, including classification (e.g., amazon polarity user satisfied), summarization (e.g., news editorial summary), translation (e.g., translate/french-english), question answering (e.g., trivia qa), and many others. Similarly, the Dolly Dataset consists of multiple instruction-following tasks across 8 categories, including question answering, classification, brainstorming, information extraction, summarization, and creative writing. In our experiments, we define tasks as the original subtasks or categories annotated in the dataset, and distribute the annotation budget across these predefined subtasks.

In modern LLM development, task and domain categorizations are widely available. When building SFT datasets, practitioners need to curate specialized prompts for each domain or application need. For example, both enterprise and academic efforts now emphasize domain-aligned SFT to improve performance in specialized settings such as legal reasoning, biomedical QA, customer support, and other specific use cases.

In this paper, we utilize these task categorizations to demonstrate that task-diverse annotation–where specific budgets are allocated to each individual task category–is as powerful as, if not more powerful than, the diversity methods mentioned previously. With a reweighting scheme based on model confidences, our method yields comparable or superior results to previous approaches. Additionally, our approach validates the current practice of curating task-specific subsets of SFT data in LLM development.

As illustrated in Figure 2, our method explicitly incorporates task information to guide selection, ensuring that all tasks are represented in the annotated subset. Previous diversity-based selection methods, which rely on embedding space coverage, often overlook task-level imbalances—leading to underrepresentation of tasks with fewer exam-

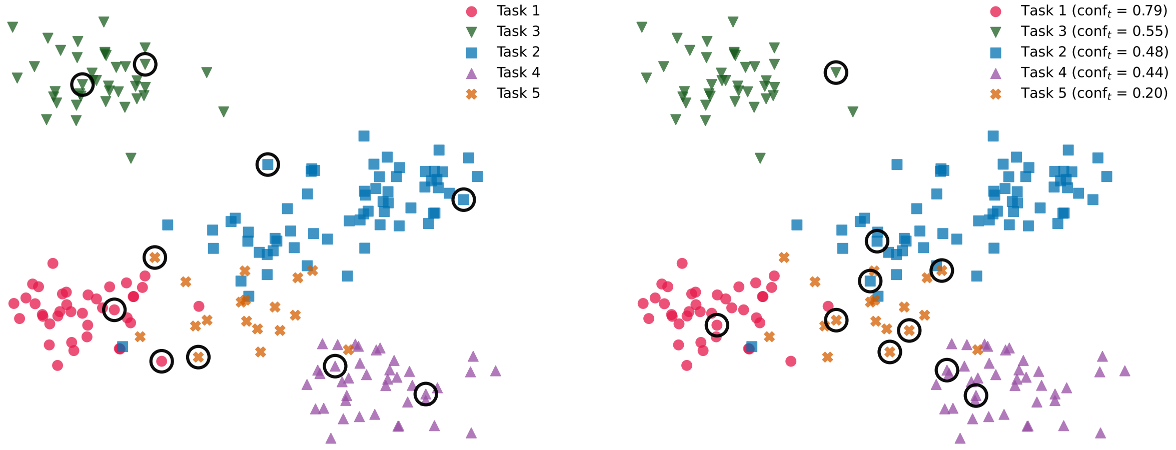


Figure 2: (a) Task diversity: Fixed number of examples are selected in a round-robin manner across tasks, without considering the model's confidence of the task. (b) Weighted Task Diversity: Examples are allocated across tasks in proportion to the inverse of the average model confidence on each task (denoted by conf_t for task t). Tasks for which the model exhibits lower confidence receive more samples. Despite Task 5 being less represented in the original distribution, we collect more samples because the pre-trained model is generally less confident about the task. When comparing to Figure 1, our method leverages the additional task information for selection.

ples or lower model confidence. In contrast, the Weighted Task Diversity strategy adapts allocation based on model confidence, assigning more examples to lower-confidence tasks, such as Task 5.

4.2 Task Diversity

As a novel baseline method that leverages task-diversity, we optimize for task-level diversity in the selected subset under a fixed annotation budget. Our objective is simple: allocate roughly equal number of examples to each task subject to the availability of prompts available in each task. We propose a two-step sampling algorithm that first allocates the budget across tasks and then samples examples from each task. This approach is denoted as **Task Diversity** in our results.

Formally, under a fixed annotation budget B, we formulate the allocation problem as minimizing the maximum number of examples per task while ensuring complete budget utilization and task-specific availability constraints. Formally, let there be T tasks with their corresponding prompt partitions $X_1, ..., X_T \subset X$. Here, X_t represents the prompts belonging to task t. We also let variables $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \dots, \alpha_T] \in \mathbb{R}^T$ denote the variable of allocated examples per task. The goal is to solve the following min-max problem:

$$\boldsymbol{\alpha}_{\text{Div}} = \arg\min_{\boldsymbol{\alpha}} \left(\max_{t \in \{1,\dots,T\}} \alpha_t \right)$$

$$\text{s.t.} \sum_{t=1}^{T} \alpha_t = B \quad \text{and} \quad \forall t \in [T], \alpha_t \leq |X_t|.$$

Here, $\boldsymbol{\alpha}_{\text{Div}} \in [0,1]^T$ denotes the budget allocations. The optimal solution would saturate the tasks with fewer amount of examples, while allocating equal amount of budget to the rest of the tasks. As elements in $\boldsymbol{\alpha}_{\text{Div}}$ may not be integers, we adopt a round robin algorithm for selection as will be described in Section 4.4.

4.3 Weighted Task Diversity

In the second method, our goal is to prioritize sampling from tasks where the model exhibits lower confidence, while still maintaining task-level coverage under a fixed annotation budget. We propose a two-step sampling algorithm that first allocates the budget across tasks based on their average model confidence and then samples examples from each task. This approach is denoted as **Weighted Task Diversity** in our results.

Under a fixed annotation budget B, we first compute the average confidence score conf_t for each task $t \in \{1, \dots, T\}$. We define the average task-level confidence conf_t as the mean model confidence over all unlabeled examples within task t, where confidence for an individual example is computed as the product of token-level probabilities in the generated sequence (Settles, 2009; Bhatt et al., 2024). Formally, for a given prompt x, let the auto-regressive base model g generate a response $y = (y_1, y_2, \dots, y_m)$, where y_j is the j-th token in the sequence. The model's confidence for this example is defined as: $\text{conf}(x) = \prod_{j=1}^{m} g(y_j \mid y_{<j}, x)$. Then, the task-level average confidence

Algorithm 1 Round-Robin Sampling

```
Input: Budget $k$ and allocation $\boldsymbol{\alpha}$.
Input: Pool of prompts $X = X_1, ..., X_T$ partitioned based on tasks $1, .., T$ and sorted based on allocation budget so that $\alpha_1 \leq \alpha_2 \leq \cdots \leq \alpha_T$.
Initialize: Selected examples: $S \leftarrow \emptyset$; counters for each task $t$: $c_t \leftarrow 0, \forall t \in [T]$.
while True do
    for t = 1,..., T do
        if $c_t < \lceil \alpha_t \rceil$ and $c_t < |X_t|$ then
            Randomly select example $x$ from $X_t$ that is not in $S$ yet.
            Update selected set: $S \leftarrow S \cup \{x\}$ and $c_t \leftarrow c_t + 1$
            if $|S| = B$ then break
        end if
    end for
end while
return Selection set $S$.
```

can be computed as an average over all prompts: $\text{conf}_t = \frac{1}{|X_t|} \sum_{x \in X_t} \text{conf}(x)$ for task t.

During our selection, each task is initially allocated a small base budget (e.g., 5 examples) to ensure minimum task coverage. The remaining budget is then distributed across tasks with more than five available examples in proportion to the inverse of their confidence scores, thus favoring tasks where the model is more uncertain. Recall that $X_1, X_2, \ldots, X_T$ denote the set of prompt corresponding to each task, and $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_T]$ denotes the final allocated samples per task. The allocation rule can be formally written as solving for $C \in \mathbb{R}^+$,

$$\alpha_t = \left[C \cdot \frac{1}{\text{conf}_t} \right]_5^{|X_t|} \quad \text{and} \quad \sum_{t=1}^{T} \alpha_t = k.$$

Where C is a normalization constant used to proportionally distribute the remaining annotation budget among eligible tasks. In addition, $[z]_a^b := \min(\max(z, a), b)$ denotes the clamping operation, ensuring the number of selected examples per task remains within valid bounds. We use $\boldsymbol{\alpha}_{\text{Weighted}}$ to denote the allocation. Overall, if all tasks have sufficient number of examples and the budget is sufficiently large, the number of examples allocated to each task is exactly proportional to its inverse confidence score.

4.4 Round Robin

As we noted before, both $\boldsymbol{\alpha}_{\text{Div}}$ and $\boldsymbol{\alpha}_{\text{Weighted}}$ may yield non-integer allocations. In practice, we round up the values $\lceil \boldsymbol{\alpha}_{\text{Div}} \rceil$ and $\lceil \boldsymbol{\alpha}_{\text{Weighted}} \rceil$, where the rounding operation is computed elementwise. To ensure we only select k examples in total, as shown in Algorithm 1, we adopt a round-robin allocation scheme that iteratively distributes the budget across tasks. We initialize all tasks as eligible for allocation and sequentially assign one example to each task in turn. This process is repeated in a loop, until either the task-specific budget is exhausted or all prompts in a task have all been selected. This procedure continues until the entire budget is fully allocated. During this process, we ensure the smaller budget tasks are prioritized in receiving examples. Within each task, we simply choose prompts uniformly without replacement.

5 Experiments

5.1 Experiment Setup

Dataset. We utilize a curated 90K subset of the FLAN V2 dataset (Longpre et al., 2023), as processed by Wang et al. (2023). FLAN V2 is an instruction finetuning dataset that integrates data from multiple sources, including FLAN 2021, P3++, Super-Natural Instructions, and a range of additional datasets focused on reasoning, dialogue, and program synthesis. In addition, we use the Databricks Dolly 15K dataset (Conover et al., 2023), which comprises 15,000 human-generated instruction-following examples. The dataset was crowd-sourced from thousands of Databricks employees, who authored prompt-response pairs across eight instruction categories

Models and Training Procedure. We conduct our experiments using the 7B parameter version of the LLaMA-2 language model (Touvron et al., 2023), considering various annotation budgets. Before finetuning, we select a subset of prompts for annotation using the baseline algorithms described below and our **Task Diversity**, **Weighted Task Diversity** algorithms. These strategies are computed based on the original, pre-trained model only. After selection, we finetune the model on the annotated prompt-response pairs using Low-rank Adaptation (Hu et al., 2021). We finetune each model for 3 epochs using the Adam optimizer with a learning rate of 10^{-4}.

Baseline Algorithms. We evaluate a diverse set of baseline algorithms for prompt selection mostly

	MMLU Score				BBH Score			
Strategy	**k = 20K**	**k = 30K**	**k = 45K**	**k = 90K**	**k = 20K**	**k = 30K**	**k = 45K**	**k = 90K**
Random	$45.67_{\pm 0.04}$	$46.28_{\pm 0.48}$	$47.85_{\pm 0.10}$	$48.94_{\pm 0.17}$	$38.42_{\pm 0.20}$	$39.99_{\pm 0.40}$	$38.14_{\pm 1.17}$	$40.02_{\pm 0.24}$
Mean Entropy	$43.79_{\pm 0.33}$	$45.49_{\pm 0.25}$	$46.82_{\pm 0.27}$	$48.94_{\pm 0.17}$	$38.45_{\pm 0.26}$	$38.27_{\pm 0.65}$	$40.14_{\pm 0.27}$	$40.02_{\pm 0.24}$
Confidence	$45.24_{\pm 0.24}$	$45.70_{\pm 0.25}$	$47.20_{\pm 0.24}$	$48.94_{\pm 0.17}$	$39.50_{\pm 0.69}$	$38.68_{\pm 0.82}$	$39.55_{\pm 0.20}$	$40.02_{\pm 0.24}$
Mean Margin	$44.99_{\pm 0.03}$	$46.18_{\pm 0.08}$	$47.22_{\pm 0.18}$	$48.94_{\pm 0.17}$	$39.40_{\pm 0.06}$	$39.65_{\pm 0.45}$	$39.86_{\pm 0.47}$	$40.02_{\pm 0.24}$
Min Margin	$46.09_{\pm 0.11}$	$46.51_{\pm 0.06}$	$47.65_{\pm 0.24}$	$48.94_{\pm 0.17}$	$\underline{40.44}_{\pm 0.48}$	$40.42_{\pm 0.42}$	$38.68_{\pm 0.48}$	$40.02_{\pm 0.24}$
k-Center	$45.83_{\pm 0.16}$	$46.47_{\pm 0.14}$	$47.74_{\pm 0.14}$	$48.94_{\pm 0.17}$	$38.65_{\pm 0.29}$	$38.94_{\pm 0.53}$	$38.32_{\pm 0.77}$	$40.02_{\pm 0.24}$
FL(γ=0.1)	$44.01_{\pm 0.23}$	$45.31_{\pm 0.92}$	$46.47_{\pm 0.17}$	$48.94_{\pm 0.17}$	$37.40_{\pm 0.13}$	$38.17_{\pm 0.68}$	$40.09_{\pm 0.83}$	$40.02_{\pm 0.24}$
FL(γ=0.002)	$44.73_{\pm 0.21}$	$46.92_{\pm 0.43}$	$47.71_{\pm 0.20}$	$48.94_{\pm 0.17}$	$38.91_{\pm 0.40}$	$40.70_{\pm 0.46}$	$\mathbf{41.68}_{\pm 0.34}$	$40.02_{\pm 0.24}$
FL(cosine)	$44.49_{\pm 0.09}$	$45.18_{\pm 0.27}$	$46.51_{\pm 0.24}$	$48.94_{\pm 0.17}$	$39.53_{\pm 0.35}$	$39.27_{\pm 1.02}$	$40.40_{\pm 0.22}$	$40.02_{\pm 0.24}$
ActiveIT	$\underline{47.50}_{\pm 0.15}$	$\underline{47.56}_{\pm 0.25}$	$47.66_{\pm 0.24}$	$48.94_{\pm 0.17}$	$39.52_{\pm 0.11}$	$39.88_{\pm 0.08}$	$40.78_{\pm 0.59}$	$40.02_{\pm 0.24}$
DPP	$45.58_{\pm 0.15}$	$46.87_{\pm 0.10}$	$47.84_{\pm 0.26}$	$48.94_{\pm 0.17}$	$\mathbf{40.68}_{\pm 0.50}$	$\underline{40.99}_{\pm 0.47}$	$40.07_{\pm 0.41}$	$40.02_{\pm 0.24}$
Task Diversity	$45.83_{\pm 0.57}$	$46.25_{\pm 0.12}$	$\mathbf{48.63}_{\pm 0.18}$	$48.94_{\pm 0.17}$	$39.02_{\pm 0.64}$	$39.63_{\pm 0.36}$	$\underline{41.10}_{\pm 0.53}$	$40.02_{\pm 0.24}$
Weighted Task Diversity	$\mathbf{47.88}_{\pm 0.19}$	$\mathbf{48.34}_{\pm 0.16}$	$\underline{48.46}_{\pm 0.15}$	$48.94_{\pm 0.17}$	$39.96_{\pm 0.52}$	$\mathbf{41.04}_{\pm 0.33}$	$40.86_{\pm 0.15}$	$40.02_{\pm 0.24}$

Table 1: Comparison of model performance on MMLU (left) and BBH (right) benchmarks using different data selection strategies and annotation budgets. Each result is averaged over 3 random seeds where the randomness mainly comes from the training. The confidence intervals are based on standard error. The best results for each k are in **bold**, and the second-best results are underlined.

adopted from Kung et al. (2023), Bhatt et al. (2024) and Wang et al. (2024). **Random** represents the random sampling baseline. We also include **Mean Entropy**, which measures the average tokenwise negative entropy of the softmax probability scores in the generated sequence; **Least Confidence**, which selects prompts with the lowest overall sequence probability, computed as the product of the token probabilities; **Mean Margin**, which captures uncertainty as the average difference between the highest and second-highest token probabilities at each position in the sequence; and **Min Margin**, a more targeted variant that considers the smallest such margin across all tokens, rather than the average. Furthermore, the task-wise allocation method **ActiveIT** (Kung et al., 2023) is also included. In this method, algorithm selects tasks where the model exhibits the lowest average confidence and annotates all examples from these tasks.

Lastly, we also include diversity-based selection methods such as **K-Center**, **Facility Location** and **DPP** (detailed in Section 2.2.1).

Evaluation Metrics. We evaluate our finetuned model's zero-shot capabilities using three benchmarks: MMLU (Massive Multitask Language Understanding, Hendrycks et al. (2020)), BBH (Big-Bench Hard, Suzgun et al. (2022)), and AlpacaEval (Li et al., 2023). Following FLAN V2's methodology, MMLU tests factual knowledge and reasoning across 57 subjects via multiple-choice questions spanning elementary to professional levels, while BBH evaluates general reasoning capabilites of the model through 23 generation-based. AlpacaEval is an evaluation framework designed to automatically measure the quality of responses generated by large language models in instruction-following tasks. It compares outputs from two models to determine which response better follows the given instruction. In our study, we used AlpacaEval 2.0 with the GPT-4 Turbo model as the annotator, which ranks the outputs from our fine-tuned model (trained on a subset of data) against those from a reference model (trained on the full dataset). For our results, we report the length controlled win rates, accounting for potential biases of long answer preferences by GPT-4 Turbo.

5.2 Evaluation on MMLU and BBH

In Table 1, when experimenting with the FLAN V2 dataset, we observe that Weighted Task Diversity substantially outperform traditional sampling methods on the MMLU benchmark under different annotation budgets. At $45K$ budget, models trained based on Weighted Task Diversity selection achieves similar performance to the model trained on all $90K$ examples in the dataset, effectively saving 50% annotation budget when compared to random sampling. Overall, this demonstrates that task-aware sampling, particularly when informed by model uncertainty, can achieve near-optimal performance even under low budgets. On the BBH benchmark, however, performance varies more significantly across strategies and budgets, with no single method consistently dominating. However, at 30K budget, our strategy achieves the best overall performance with a score surpassing the accuracy

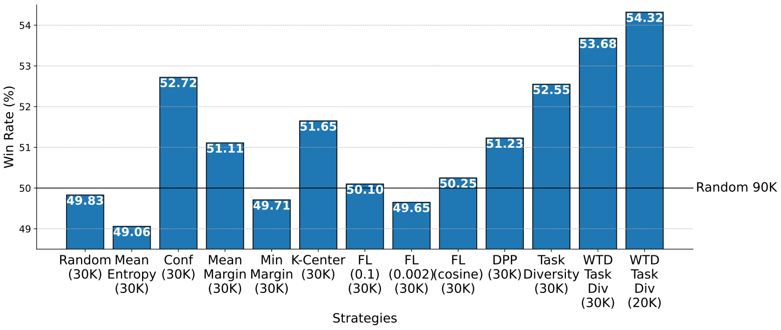


Figure 3: Evaluation of 30K prompt selection strategies on the FLAN V2 dataset using AlpacaEval. The win rate represents the proportion of times each model's response was preferred by GPT-4 Turbo over the 90K random baseline. "WTD Task Div" here indicates Weighted Task Diversity and "Conf" indicates Confidence

Strategy	**k = 3K**	**k = 6K**	**k = 13.5K**
Random	$33.33_{\pm0.49}$	$34.65_{\pm0.84}$	$35.33_{\pm1.03}$
Mean Entropy	$\underline{34.35}_{\pm0.31}$	$33.74_{\pm0.76}$	$35.33_{\pm1.03}$
Confidence	$34.29_{\pm1.71}$	$37.51_{\pm0.60}$	$35.33_{\pm1.03}$
Mean Margin	$32.61_{\pm1.96}$	$33.01_{\pm0.74}$	$35.33_{\pm1.03}$
Min Margin	$29.43_{\pm0.35}$	$33.85_{\pm1.22}$	$35.33_{\pm1.03}$
k-Center	$33.61_{\pm0.85}$	$32.90_{\pm1.17}$	$35.33_{\pm1.03}$
FL(γ = 0.1)	$33.19_{\pm2.38}$	$33.11_{\pm0.15}$	$35.33_{\pm1.03}$
FL(γ = 0.002)	$31.70_{\pm1.73}$	$33.22_{\pm1.76}$	$35.33_{\pm1.03}$
FL(cosine)	$34.03_{\pm1.97}$	$\underline{35.52}_{\pm0.66}$	$35.33_{\pm1.03}$
DPP	$31.02_{\pm1.49}$	$32.70_{\pm1.29}$	$35.33_{\pm1.03}$
Task diversity	$33.62_{\pm0.34}$	$32.65_{\pm0.42}$	$35.33_{\pm1.03}$
Weighted Task Diversity	$\mathbf{39.74}_{\pm0.54}$	$\mathbf{38.93}_{\pm0.34}$	$35.33_{\pm1.03}$

Table 2: MMLU evaluation of models trained on subsets selected from a pool of 13.5K examples from the Dolly dataset. Each result is averaged over 3 trials.

when labeling all $90K$ examples.

In Table 2, we also include the performance of Weighted Task Diversity on the Dolly dataset. Our method achieves the highest MMLU scores at both 3K and 6K budgets, reaching 39.74% and 38.93% accuracies, respectively. Notably, as we are selecting a more informative set and balanced set of examples, the MMLU score even outperforms training on all examples by more than 4% and saving 80% in annotation budget. We also include the BBH result for the Dolly dataset in Table 3 of Appendix A. However, even with random sampling, we see the model performance consistently dropping as more examples are being labeled and trained on. This potentially suggests the ineffectiveness of the Dolly dataset in improving the tasks covered by BBH, suggesting us to downweight the significance of this benchmark.

5.3 Evaluation by GPT-4

To further investigate the effectiveness of various data selection strategies under constrained annotation budgets, we use the finetuned models on 30K prompts for all strategies and compare them against a baseline model trained on 90K randomly selected prompts. Evaluation is conducted using GPT-4 Turbo as the judge following the AlpacaEval framework (Li et al., 2023), having over 805 prompts. As shown in Figure 3, several strategies surpass the Random (90K) baseline, despite using significantly fewer examples.

Among all strategies, Weighted Task Diversity 20K and 30K budget achieve the highest win rates of 54.32% and 53.68% for FLAN Datset. Even on the Dolly Dataset as shown in Figure 5, Weighted Task Diversity with 3K and 6K budgets achieves the highest win rates of 52.28% and 51.17%, respectively, outperforming all other strategies.

Overall, these evidences suggest that the simple method of Weighted Task Diversity is comparable if not better than the existing baseline methods.

5.4 Qualitative Analysis

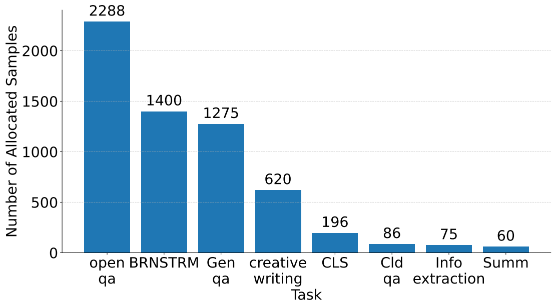


Figure 4: Number of allocated samples per task in the Dolly dataset under the Weighted Task Diversity strategy. "BRNSTRM" indicates Brainstorming Tasks, "CLS" indicates Classification Tasks, "Cld qa" indicates Closed QA and "Summ" indicates Summarization.

In Figure 4, we plot the task-wise allocation budget of the **Weighted Task Diversity** strategy for the Dolly dataset. As we can observe, a majority of samples are allocated to open QA, brainstorming, and general QA tasks. This distribution aligns with intuitive expectations: tasks like open QA typically present greater challenges for pre-trained language models due to their open-ended nature and broad variability in possible responses. The relatively lower model confidence on QA-like tasks leads to greater sampling from these categories, consistent with the inverse-confidence weighting mechanism. In contrast, tasks such as classification, summarization, and information extraction receive far fewer samples, likely because they are structurally simpler and the model exhibits higher confidence on them. However, unlike the method proposed by Kung et al. (2023), these tasks still receive a few annotations, ensuring our dataset has a broad coverage over them. See Appendix B for more information on the FLAN dataset.

6 Conclusion

In this paper, we present a simple yet effective approach to label-efficient supervised finetuning by leveraging task-level diversity and model confidence. We introduce two algorithms–Task Diversity and Weighted Task Diversity–that allocate annotation budgets across tasks based on either uniform distribution or inverse model confidence scores. Our methods consistently outperform or match more complex diversity- and uncertainty-based baselines across MMLU and GPT-4-based AlpacaEval benchmarks, all while using significantly fewer labeled examples. These results highlight the value of leveraging task structure and model uncertainty for cost-effective and scalable instruction tuning of large language models.

Limitations

Our approach demonstrates strong performance in label-efficient learning scenarios, though we acknowledge several areas for future exploration. While task information is widely available in modern instruction-tuning datasets as we've shown, there may be specific domains where such categorizations are less defined. In these cases, automated methods for identifying task definitions and categorizations could be beneficial.

The confidence assessment by the base model may sometimes reflect its pre-training exposure rather than inherent task difficulty. However, our experimental results suggest this concern is largely mitigated by our round-robin sampling strategy that ensures minimum coverage across all tasks.

Our experiments primarily focus on the LLaMA-2 7B architecture, and extending these findings to other model architectures represents a promising direction for future work. Additionally, while our method is significantly more computationally efficient than existing diversity-based approaches, calculating confidence scores does require inference on the unlabeled dataset.

Acknowledgments

This work has been supported in part by NSF Award 2112471.

References

Jordan Ash, Surbhi Goel, Akshay Krishnamurthy, and Sham Kakade. 2021. Gone fishing: Neural active learning with fisher embeddings. *Advances in Neural Information Processing Systems*, 34:8927–8939.

Jordan T Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. 2019. Deep batch active learning by diverse, uncertain gradient lower bounds. *arXiv preprint arXiv:1906.03671*.

Gantavya Bhatt, Yifang Chen, Arnav M Das, Jifan Zhang, Sang T Truong, Stephen Mussmann, Yinglun Zhu, Jeffrey Bilmes, Simon S Du, Kevin Jamieson, Jordan T Ash, and Robert D Nowak. 2024. An experimental design framework for label-efficient supervised finetuning of large language models. *arXiv preprint arXiv:2401.06692*.

Jeff Bilmes. 2022. Submodularity in machine learning and artificial intelligence. *arXiv preprint arXiv:2202.00132*.

Alexander Bukharin and Tuo Zhao. 2023. Data diversity matters for robust instruction tuning. *Preprint*, arXiv:2311.14736.

Hardy Chen, Haoqin Tu, Fali Wang, Hui Liu, Xianfeng Tang, Xinya Du, Yuyin Zhou, and Cihang Xie. 2025. Sft or rl? an early investigation into training r1-like reasoning large vision-language models. *arXiv preprint arXiv:2504.11468*.

Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, and Hongxia Jin. 2023. Alpagasus: Training a better alpaca with fewer data. *Preprint*, arXiv:2307.08701.

Gui Citovsky, Giulia DeSalvo, Claudio Gentile, Lazaros Karydas, Anand Rajagopalan, Afshin Rostamizadeh, and Sanjiv Kumar. 2021. Batch active learning at scale. *Advances in Neural Information Processing Systems*, 34:11933–11944.

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. 2023. Free dolly: Introducing the world's first truly open instruction-tuned llm.

Qianlong Du, Chengqing Zong, and Jiajun Zhang. 2023. Mods: Model-oriented data selection for instruction tuning. *Preprint*, arXiv:2311.15653.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Hamish Ivison, Noah A. Smith, Hannaneh Hajishirzi, and Pradeep Dasigi. 2022. Data-efficient fine-tuning using cross-task nearest neighbors. arXiv preprint arXiv:2212.00196.

Jan Kremer, Kim Steenstrup Pedersen, and Christian Igel. 2014. Active learning with support vector machines. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 4(4):313–326.

Alex Kulesza, Ben Taskar, and 1 others. 2012. Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning*, 5(2–3):123–286.

Po-Nien Kung, Fan Yin, Di Wu, Kai-Wei Chang, and Nanyun Peng. 2023. Active instruction tuning: Improving cross-task generalization by training on prompt sensitive tasks. *arXiv preprint arXiv:2311.00288*.

David D Lewis. 1995. A sequential algorithm for training text classifiers: Corrigendum and additional data. In *Acm Sigir Forum*, volume 29, pages 13–19. ACM New York, NY, USA.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Alpacaeval: An automatic evaluator of instruction-following models. https://github.com/tatsu-lab/alpaca_eval.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, and 1 others. 2023. The flan collection: Designing data and methods for effective instruction tuning. *arXiv preprint arXiv:2301.13688*.

Pitu B Mirchandani and Richard L Francis. 1990. *Discrete location theory*.

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

Mohamad Amin Mohamadi, Wonho Bae, and Danica J Sutherland. 2022. Making look-ahead active learning strategies feasible with neural tangent kernels. *Advances in Neural Information Processing Systems*, 35:12542–12553.

Shyam Nuggehalli, Jifan Zhang, Lalit Jain, and Robert Nowak. 2023. Direct: Deep active learning under imbalance and label noise. *arXiv preprint arXiv:2312.09196*.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. *Preprint*, arXiv:2203.02155.

Yulei Qin, Yuncheng Yang, Pengcheng Guo, Gang Li, Hang Shao, Yuchen Shi, Zihan Xu, Yun Gu, Ke Li, and Xing Sun. 2024. Unleashing the power of data tsunami: A comprehensive survey on data assessment and selection for instruction tuning of language models. *arXiv preprint arXiv:2408.02085*.

Ozan Sener and Silvio Savarese. 2018. Active learning for convolutional neural networks: A core-set approach. In *International Conference on Learning Representations (ICLR)*.

Burr Settles. 2009. Active learning literature survey.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, and 1 others. 2022. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*.

Simon Tong and Daphne Koller. 2001. Support vector machine active learning with applications to text classification. *Journal of machine learning research*, 2(Nov):45–66.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Peiqi Wang, Yikang Shen, Zhen Guo, Matthew Stallone, Yoon Kim, Polina Golland, and Rameswar Panda. 2024. Diversity measurement and subset selection for instruction tuning datasets. *arXiv preprint arXiv:2402.02318*.

Yizhong Wang, Hamish Ivison, Pradeep Dasigi, Jack Hessel, Tushar Khot, Khyathi Raghavi Chandu, David Wadden, Kelsey MacMillan, Noah A Smith, Iz Beltagy, and 1 others. 2023. How far can camels go? exploring the state of instruction tuning on open resources. *arXiv preprint arXiv:2306.04751*.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Gary Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Maitreya Patel, and 21 others. 2022. Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks. *Preprint*, arXiv:2204.07705.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022. Finetuned language models are zero-shot learners. *Preprint*, arXiv:2109.01652.

Kai Wei, Rishabh Iyer, and Jeff Bilmes. 2015. Submodularity in data subset selection and active learning. In *International conference on machine learning*, pages 1954–1963. PMLR.

Tian Xie, Jifan Zhang, Haoyue Bai, and Robert Nowak. 2024. Deep active learning in the open world. *arXiv preprint arXiv:2411.06353*.

Yu Yang, Siddhartha Mishra, Jeffrey Chiang, and Baharan Mirzasoleiman. 2024. Smalltolarge (s2l): Scalable data selection for fine-tuning large language models by summarizing training trajectories of small models. *Advances in Neural Information Processing Systems*, 37:83465–83496.

Jifan Zhang, Gregory Canal, Yinglun Zhu, Robert D Nowak, Yifang Chen, Arnav M Das, Gantavya Bhatt, Stephen Mussmann, Jeffrey Bilmes, Simon S Du, and 1 others. 2023. Labelbench: A comprehensive framework for benchmarking adaptive label-efficient learning. *arXiv preprint arXiv:2306.09910*.

Jifan Zhang, Julian Katz-Samuels, and Robert Nowak. 2022. Galaxy: Graph-based active learning at the extreme. In *International Conference on Machine Learning*, pages 26223–26238. PMLR.

Kuan Lok Zhou, Jiayi Chen, Siddharth Suresh, Reuben Narad, Timothy T Rogers, Lalit K Jain, Robert D Nowak, Bob Mankoff, and Jifan Zhang. 2025. Bridging the creativity understanding gap: Small-scale human alignment enables expert-level humor ranking in llms. *arXiv preprint arXiv:2502.20356*.

A Additional Results

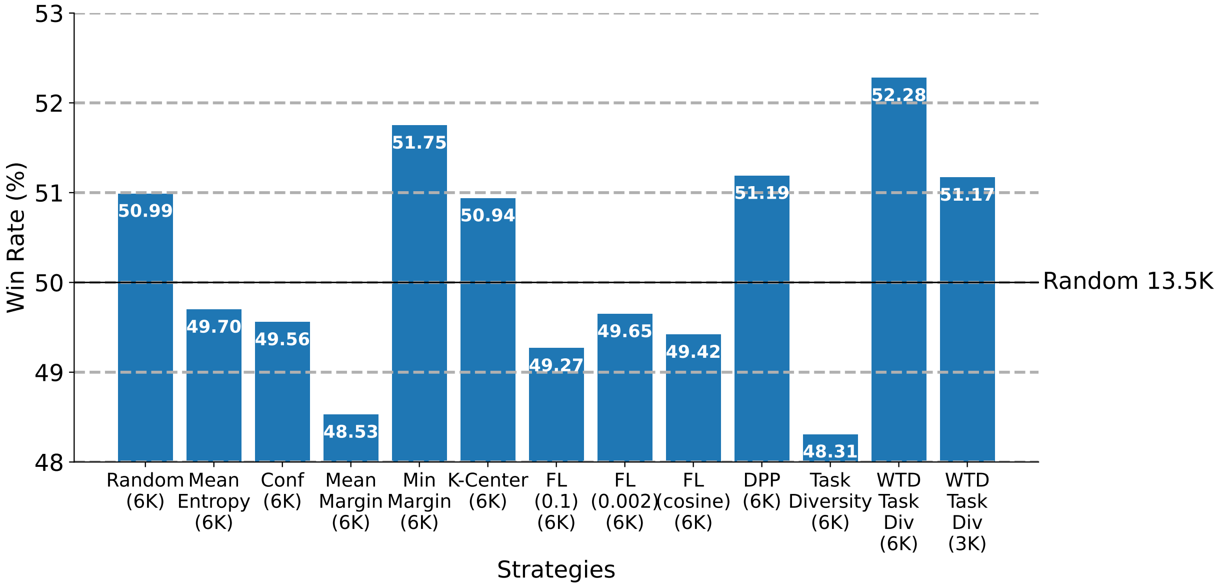


Figure 5: Evaluation of 6K prompt selection strategies on the Dolly dataset using AlpacaEval. The win rate represents the proportion of times each model's response was preferred by GPT-4 Turbo over the 13.5K random baseline. "WTD Task Div" here indicates Weighted Task Diversity and "Conf" indicates Confidence

Strategy	**k = 3K**	**k = 6K**	**k = 13.5K**
Random	$36.43_{\pm 0.11}$	$35.60_{\pm 0.72}$	$34.02_{\pm 0.84}$
Mean Entropy	$35.48_{\pm 0.79}$	$34.07_{\pm 0.36}$	$34.02_{\pm 0.84}$
Confidence	$36.63_{\pm 0.40}$	$35.68_{\pm 0.55}$	$34.02_{\pm 0.84}$
Mean Margin	$35.79_{\pm 0.71}$	$35.99_{\pm 0.37}$	$34.02_{\pm 0.84}$
Min Margin	$36.32_{\pm 0.28}$	$33.89_{\pm 1.02}$	$34.02_{\pm 0.84}$
k-Center	$\mathbf{38.99}_{\pm 0.45}$	$35.43_{\pm 0.65}$	$34.02_{\pm 0.84}$
FL($\gamma = 0.1$)	$35.76_{\pm 0.28}$	$\mathbf{37.12}_{\pm 0.24}$	$34.02_{\pm 0.84}$
FL($\gamma = 0.002$)	$35.84_{\pm 0.20}$	$35.89_{\pm 0.62}$	$34.02_{\pm 0.84}$
FL(cosine)	$34.84_{\pm 0.90}$	$35.48_{\pm 0.02}$	$34.02_{\pm 0.84}$
DPP	$\underline{37.27}_{\pm 0.27}$	$\underline{36.38}_{\pm 0.30}$	$34.02_{\pm 0.84}$
Task diversity	$34.96_{\pm 0.54}$	$35.60_{\pm 0.38}$	$34.02_{\pm 0.84}$
Weighted task diversity	$37.07_{\pm 0.30}$	$35.63_{\pm 0.45}$	$34.02_{\pm 0.84}$

Table 3: Big Bench Hard (BBH) evaluation of models trained on subsets selected by strategies from a pool of 13.5K under different annotation budgets on the Dolly dataset. Each result is averaged over 3 random seeds.

B Qualitative Analysis

In Figure 6, we visualize the task-wise allocation under the **Weighted Task Diversity** strategy for the FLAN dataset. A significant portion of the annotation budget is allocated to math-related tasks such as math dataset algebra, where pre-trained language models tend to exhibit high uncertainty. Moderate allocation is also seen for paraphrase detection tasks (e.g., Quora Question Pairs(QQP) and MNLI(Multi-Genre Natural Language Inference), which often require nuanced semantic understanding. In contrast, although we set a base allocation of 5 examples per task, some tasks such as prachathai67k sentiment classification and low-resource translation contain only a single available example and are selected accordingly. This behavior reflects the influence of task size constraints rather than model confidence.

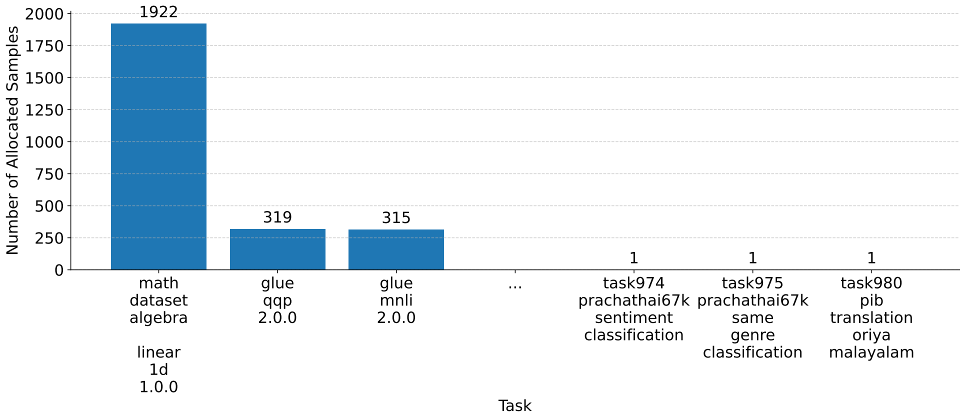


Figure 6: Task-wise sample allocation in the **FLAN V2** dataset using the **Weighted Task Diversity** strategy. Tasks where the model exhibits greater uncertainty receive a larger portion of the annotation budget.

Nevertheless, such tasks still receive limited annotations, helping to preserve broad task-level coverage and ensuring that all tasks are represented in the final training set.

C Additional Experiment Details

Licenses for Models and Datasets

The following licenses apply to the models and datasets used in this work:

- **Llama-2 7B**: Released by Meta under a custom non-commercial license.[1]
- **FLAN V2 Dataset**: Released by Google under the Apache License 2.0.[2]
- **Dolly Dataset**: Released by Databricks under the Creative Commons Attribution-ShareAlike 3.0 (CC BY-SA 3.0) license.[3]

Computational Complexity

We primarily used L40 GPUs. We precompute the embeddings which takes 12 hours. For FLAN Dataset using L40, each trial takes roughly 35 GPU hours including the evaluation for 90K budget, 23 GPU hours for 45K, 18 hours for 30K and 15 GPU hours for 20K respectively. For Dolly Dataset using L40, each trial takes roughly 10 GPU hours including the evaluation for 13.5K budget, 9 GPU hours for 6K, 8 GPU hours for 3K respectively.

[1] https://ai.meta.com/resources/models-and-libraries/llama-downloads/

[2] https://github.com/google-research/FLAN/tree/main/flan/v2

[3] https://github.com/databrickslabs/dolly