EffiEval: Efficient Model Evaluation via Capability Coverage Maximization

Anonymous ACL submission

Abstract

The rapid advancement of large language models (LLMs) and the development of increasingly large and diverse evaluation benchmarks have introduced substantial computational challenges for model assessment. In this paper, we present EffiEval, a training-free approach for efficient benchmarking that effectively addresses data redundancy while maintaining high evaluation reliability. Our method is specifically designed to meet three key criteria for high-quality evaluation: **representativeness**, by ensuring comprehensive coverage of model capabilities; **fairness**, by remaining independent of model performance during sample selection to avoid bias; and **generalizability**, by enabling flexible transfer across datasets and model families without reliance on large-scale evaluation data. Unlike traditional methods that rely on absolute performance or require extensive evaluation data, our approach adaptively selects high-quality representative subsets based on the Model Utility Index (MUI). Extensive experiments on multiple public benchmarks and diverse LLMs demonstrate that EffiEval achieves strong ranking consistency with full-dataset evaluation using only a small fraction of the original data. Furthermore, our method is flexible and scalable in size, allowing users to balance evaluation efficiency and representativeness according to specific needs. Overall, EffiEval provides a practical and generalizable solution for reliable, fair, and efficient evaluation in the era of LLMs.

1 Introduction

Based on the training scaling laws (Kaplan et al., 2020), large language models (LLMs) are becoming significantly more capable as computational resources, model parameters, and training data scale up continuously. This trend has driven researchers to construct increasingly large and diverse benchmarks for comprehensive model evaluation, such as MMLU (Hendrycks et al., 2020), HELM (Liang et al., 2022), and BIG-Bench (Srivastava et al., 2023). However, the scale of these benchmarks introduces substantial evaluation costs. For example, on the HELM benchmark, evaluating a single model can require over 500 GPU hours (Liang et al., 2022). This computational burden is further increased by the growing adoption of test-time scaling (OpenAI, 2024; Guo et al., 2025), where longer inference times are used to boost performance. Worse still, the rapid iteration and frequent updating of LLMs further intensify this evaluation cost. Therefore, improving evaluation efficiency while ensuring high quality has become an increasingly important challenge in the era of LLMs.

We assume that traditional evaluation datasets contain a certain degree of redundancy. Therefore, it is possible to down-sample the data to achieve efficient benchmarking — that is, to intelligently reduce the computational cost of evaluation without compromising its reliability (Perlitz et al., 2023). This reliability can be defined in two ways: 1) the absolute value of performance metrics remains the same, which is emphasized in previous works (Polo et al., 2024; Kipnis et al., 2024), or 2) the ranking of multiple models is preserved, which we prefer because preserving the absolute values of performance metrics is unnecessary. First, absolute scores are more sensitive to data distribution shifts. Suppose a model performs particularly well on coding questions, which make up 70% of the dataset. If the proportion of coding questions decreases, the model's absolute performance score will naturally decline. However, such changes do not necessarily indicate that the selected subset is of poor quality — it may simply reflect a shift in the data distribution. Second, scores are affected by the difficulty of the sample. When easy questions are removed and the difficulty gap is widened, absolute scores will inevitably drop. Therefore, we adopt the relative ranking among multiple models as the primary measure of this new task setting.

To accomplish the task of efficient benchmarking, we highlight that a high-quality subset should also meet additional criteria to ensure that it remains representative, fair, and generalizable. Specifically: 1) It should still cover the diverse capabilities of models as much as possible to ensure comprehensive evaluation. For example, (Perlitz et al., 2023) apply stratified random sampling based on the scenarios defined in the original benchmark. However, this approach is heavily influenced by the original data distribution, making it difficult to adequately sample from sparse categories or domains, which may even be entirely missed during sampling. 2) It should be uncorrelated with model performance, to avoid introducing bias. Clearly, when the sampling process is correlated with model performance — such as selecting samples where models differ the most — it may introduce evaluation bias. 3) It should exhibit a certain level of generalizability. For example, if generating the subset requires evaluating all data on all evaluated models beforehand and cannot adapt to new model evaluations, then it does not truly improve evaluation efficiency. Previous statistic-based approaches (Polo et al., 2024; Kipnis et al., 2024) rely on large amounts of evaluation data to select informative samples, making them difficult to transfer to new datasets or adapt to unseen evaluation settings.

In this paper, we propose a training-free efficient benchmarking method, **EffiEval**, which satisfies the above three criteria. Previous work (Cao et al., 2025b; Pan et al., 2024; Templeton et al., 2024) has pointed out that different neurons in a model reflect distinct capabilities; specifically, the work in (Cao et al., 2025b) demonstrates that evaluating more capabilities activates a larger number of neurons. Inspired by this, our core idea is to reduce data redundancy by maximizing the number of activated neurons through the model's internal mechanism, while simultaneously preserving the diversity of covered capabilities. Unlike traditional diversity-based criteria (e.g., domains or predefined capabilities), our method is model-specific — it selects evaluation samples that are diverse with respect to a given model, ensuring a broad and representative assessment. To mitigate the performance bias issue, our approach does not rely on large-scale evaluation data to train sample representations or performance predictors. This ensures that the selection process remains independent of the model's actual performance, while also being efficient and easily transferable to new datasets. In later experiments, we demonstrate the generalizability of our selected subsets: subsets chosen based on one model can also provide reliable evaluation results for other models. We argue that, despite differences in training data and architectures, many models share similar distributions of capability diversity, which contributes to the observed generalization. In this sense, our approach can also be viewed as a form of meta-evaluation, enabling a quantitative assessment of dataset redundancy or diversity.

In our experiments, we observe the following findings: 1) Using as little as 5% of the original data, our method achieves an average Kendall's τ greater than 0.9 across multiple benchmarks, indicating strong preservation of evaluation rankings; 2) When increasing the subset to 10%, the average Kendall's τ exceeds 0.95, reflecting even stronger consistency with full-data evaluation; 3) Unlike prior approaches that require predefining the subset size or searching across all possible sizes, our method dynamically determines the subset size based on the desired coverage or performance correlation.

Our contributions can be summarized as follows:

- We highlight the task of efficient benchmarking and argue that a high-quality subset must meet several key requirements to ensure representativeness, fairness, and generalizability.
- We propose a training-free subset sampling method EffiEval that balances evaluation efficiency and data representativeness, without relying on large-scale evaluation data, thus enabling easy transferability to other datasets.
- Extensive experiments demonstrate that our method can adaptively select a representative subset that not only covers diverse model capabilities but also preserves the performance ranking among models.

2 Related Work

Efficient and Generalizable Evaluation. As LLM capabilities grow rapidly, fixed test datasets and static metrics can no longer keep up (Cao et al., 2025a). This growing mismatch calls for more generalizable and adaptive evaluation methods. One common direction takes the evaluator perspective, where LLMs themselves are leveraged as evaluators to reduce the cost of data construction and annotation, and to enable more up-to-date and scalable evaluation (Bai et al., 2023b; Ying et al.,

2024a). Another line of work focuses on estimating model performance based on low-cost proxies, such as model size and training token count (i.e., scaling laws (Kaplan et al., 2020)), or performance on a carefully selected, representative subset of data (Polo et al., 2024; Pacchiardi et al., 2024; Kipnis et al., 2024)—the latter being a form of evaluation data selection.

Evaluation Data Selection. Evaluating models on large benchmarks is time- and resource-intensive. To address this, several studies have proposed selecting a representative subset for evaluation and extrapolating the full dataset performance. For example, (Vivek et al., 2023) leverage confidence scores on classification benchmarks to select samples with the highest correlation in scores with the rest of the dataset. (Polo et al., 2024) fit an Item Response Theory (IRT) model using prior LLM performance, then apply K-Means clustering on the estimated item parameters to select representative samples. (Pacchiardi et al., 2024) follow a similar pipeline but use sample embeddings obtained from the OpenAI API. (Kipnis et al., 2024) also adopts an IRT-based approach but uses Fisher Information to filter out less discriminative samples. These methods rely on extensive performance data (from 400 to 5000 models on a single dataset), causing substantial computational overhead and potential bias. In contrast, our method requires minimal evaluation data yet still generalizes well, achieving high correlation between subset and full-dataset performance.

3 MUI Based Evaluation Data Selection

3.1 MUI Computation

Inspired by prior efforts to quantify model efficiency, the Model Utility Index (MUI) (Cao et al., 2025b) is proposed as the foundation of our work to measure the amount of effort a model expends to achieve a given outcome. The MUI is defined as:

$$\mathbf{MUI}(t) = \frac{N_{\text{activated}}(t)}{N_{\text{total}}} \tag{1}$$

where N_{total} denotes the total number of capabilities (e.g., neurons or features) of the model, and $N_{\text{activated}}(t)$ represents the number of activated capabilities when the model completes task t. Built upon interpretation techniques, MUI can naturally measure the extent to which a model's capabilities are exercised by specific tasks, as studies (Pan et al., 2024; Templeton et al., 2024) have shown that different types of knowledge and abilities are associated with distinct sets of key neurons or features. To ensure broader applicability, in this work, we adopt a neuron-based MUI calculation for all experiments. Specifically, Given an input sample x and its corresponding model prediction $y = (\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_t)$, Cao et al. (2025b) defines the neuron-based contribution score $f_{\text{neuron}}(i, l, \hat{y}_j \mid x) \in \mathbb{R}$ of the i-th neuron in layer l to the prediction of token $\hat{y}_j$ as

$$f_{\text{neuron}}(i, l, \hat{y}_j \mid x) = \left(\mathbf{W}_u \mathbf{W}^l_{\text{out}} \circ \sigma\left(\mathbf{W}^l_{\text{in}}\left(\mathbf{x}^l_{-1}\right)\right)^\top\right)_{i,\hat{y}_j}, \tag{2}$$

where σ is an activation function, $\mathbf{W}^l_{\text{in}}$ and $\mathbf{W}^l_{\text{out}}$ are the input/output projections in FFN, $\mathbf{W}_u$ is the unembedding matrix transforming the hidden states into scores over the vocabulary, $\circ$ is an element-wise product with broadcasting, and $\mathbf{x}^l_{-1}$ denotes the input of FFN in the last token before predicting $\hat{y}$ at l-th layer. For a given threshold η, the key activated neurons for task sample $t = (x, y)$ is defined as:

$$N_{\text{activated}}(t) = \left\{(i,l) \;\middle|\; \exists \hat{y}_j \in y_i, f_{\text{neuron}}\left(i, l, \hat{y}_j \mid x \oplus \hat{y}_{<j}\right) > \eta\right\}, \tag{3}$$

Where: $\hat{y}_{<j} = (\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_{j-1})$ denotes the partial response sequence before the j-th token $\hat{y}_j$, $l = 1, 2, ..., L$ represents the layer index, and $i = 1, 2, ..., N$ indicates the neuron index in each layer. Thereby, MUI in Eq. (1) can be revised as

$$\mathbf{MUI}_{\text{neuron}}(t) = \frac{|N_{\text{activated}}(t)|}{NL} \tag{4}$$

These definitions can be naturally extended to multiple samples $\mathcal{T} = \{t_1, t_2, \ldots, t_K\}$ as:

$$\mathbf{MUI}_{\text{neuron}}(\mathcal{T}) = \frac{|\bigcup_{i=1}^{K} N_{\text{activated}}(t_i)|}{NL} \tag{5}$$

3.2 EffiEval

Using MUI as a guiding signal, our objective is to select a subset of k representative samples $S = \{t_{i_1}, \ldots, t_{i_k}\}$ from the K-sized full dataset $\mathcal{T} = \{t_i\}_{i=1}^K$ (where $k < K$), such that the selected samples collectively maximize the coverage of the model's capabilities. This objective is equivalent to maximizing the total MUI over the selected subset:

$$\begin{aligned} S &= \arg\max_{S \subseteq \mathcal{T}} \mathbf{MUI}_{\text{neuron}}(S) \\ &= \arg\max_{S \subseteq \mathcal{T}} \left| \bigcup_{t \in S} N_{\text{activated}}(t) \right| \end{aligned} \tag{6}$$

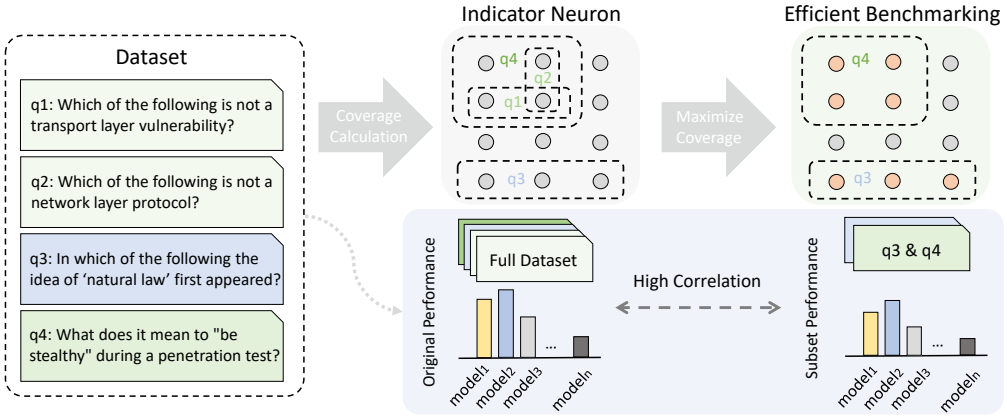

Figure 1: Framework of our method **EffiEval**. By computing the evaluation coverage of the indicator model, we select samples that maximize the set of covered capabilities, thereby constructing a representative evaluation subset as a substitute for the full benchmark. To maximize the coverage of model capabilities, we select a subset that activates the broadest range of indicator neurons. In this example, $\{q_3, q_4\}$ is selected due to its diverse neuron activation patterns. The detailed algorithm is in Algorithm 1.

Given that the calculation of MUI depends on the specific model, we refer to the model used for this calculation as the indicator. Owing to shared capabilities across models, the resulting coverage has strong generalizability: the evaluated capabilities covered by the indicator are also tested when using the representative subset on other models. Detailed quantitative analysis and further discussion of this generalization are provided in the experiments.

Algorithm 1 MUI Data Selection

Input: Key neuron set of original dataset $\{N_{\text{activated}}(t_i)\}_{i=1}^{K}$
Parameter: Subset size k
Output: Selected samples $S = \{t_{i_1}, ..., t_{i_k}\}$

Initialize $S \leftarrow \emptyset$, $N_{\text{covered}} \leftarrow \emptyset$
for $t = 1$ **to** k **do**
  Select $t_{i^*} = \arg\max_{t_i \notin S} |N_{\text{covered}} \cup N_{\text{activated}}(t_i)|$
  $S \leftarrow S \cup \{t_{i^*}\}$
  $N_{\text{covered}} \leftarrow N_{\text{covered}} \cup N_{\text{activated}}(t_{i^*})$
end for
return S

Given the optimization objective in Eq. (6), this problem can be formulated as a Maximum Coverage Problem (MCP). Although NP-Hard, it admits an efficient greedy algorithm that iteratively selects the element providing the largest marginal gain. Despite its simplicity, the greedy approach with random sampling guarantees a $(1 - 1/e)$ approximation ratio (Nemhauser et al., 1978). In this work, we adopt this method to solve the maximum coverage problem, as illustrated in Algorithm 1. The overall process of EffiEval is shown in Figure 1. Suppose we are given a full dataset $\mathcal{T} = \{q_1, q_2, q_3, q_4\}$. Each question $q_i \in \mathcal{T}$ is first mapped to its corresponding set of activated indicator neurons $N_{\text{activated}}(q_i)$, which represent the latent capabilities of the model triggered by that sample. For instance, q_1, q_2, and q_4 (from the network security domain) activate overlapping neurons, while q_3 (from the legal domain) activates a distinct region. Our goal is to select a subset S that maximizes the total MUI, i.e., the union of activated neurons across selected samples. Under this objective, the subset $\{q_3, q_4\}$ achieves broader neuron coverage and thus better represents the model's full capability spectrum.

4 Experiments

To comprehensively evaluate the effectiveness of our selected subsets as representatives of the full datasets, we conduct extensive experiments across four widely used benchmarks and 17 models, including both open-source and proprietary models.

4.1 Experiment Setting

Baselines. We compare the following baselines:

- **Random Selection**, where the samples are randomly selected from the full dataset.
- **Representation-Based Clustering**, where representative samples are selected based on clustering results. Specifically, we compute question representation embeddings using `text-embedding-3-large`, and then perform k-means clustering to group the questions, following (Pacchiardi et al., 2024) (marked as K-Means).

Method	GSM8K (k = 100)			ARC (k = 100)			Hellaswag (k = 100)			MMLU (k = 100)		
	r_S	r_K	MAE ↓	r_S	r_K	MAE ↓	r_S	r_K	MAE ↓	r_S	r_K	MAE ↓
Random	95.3	87.4	2.88	95.4	86.5	2.88	97.8	91.0	3.35	95.7	85.8	3.59
K-Means	95.0	87.0	2.76	95.8	87.2	2.78	98.1	91.5	3.30	95.8	86.5	4.59
tinyBenchmarks	89.5	79.6	**2.12**	95.4	85.1	3.29	**98.3**	91.2	6.78	96.8	87.8	**2.95**
EffiEval	**99.2**	**95.9**	4.07	**96.0**	**87.4**	**2.27**	**98.3**†	**92.5**†	**3.09**†	**96.9**	**89.1**	3.45

Table 1: Comparison between EffiEval and tinyBenchmarks in terms of 1) correlation (r_S, r_K) between the evaluated model performances on the selected subset and those on the full dataset, and 2) MAE. Entries marked with a dagger (†) indicate results obtained using `Qwen2.5-7B-Instruct`, which is used in place of LLaMA due to safety restrictions that significantly degrade LLaMA's performance. Our method outperforms the baselines with higher correlation and lower MAE. The best results are highlighted in bold, and the second-best results are underlined.

Method	GSM8K (k = 237)			ARC (k = 145)			Hellaswag (k = 93)			MMLU (k = 96)		
	r_S	r_K	MAE ↓	r_S	r_K	MAE ↓	r_S	r_K	MAE ↓	r_S	r_K	MAE ↓
Random	97.4	91.3	**1.75**	95.4	85.8	**2.17**	97.8	91.1	3.41	95.5	85.7	4.71
K-Means	98.0	91.4	**1.75**	95.5	85.8	2.50	98.2	91.9	3.49	95.8	86.5	4.74
metabench	97.0	89.3	2.30	**98.8**	**93.7**	3.27	96.5	85.5	**3.02**	**98.5**	**93.3**	6.59
EffiEval	**98.5**	**93.7**	3.62	97.1	87.8	2.77	**98.4**†	**92.8**†	3.29†	97.1	88.5	**3.63**

Table 2: Comparison between EffiEval and metabench. The evaluation setting is consistent with Table 1, except that the subset size k for each dataset is set to match the value used in `metabench`.

- **SOTA methods**: for other publicly available selection methods, such as tinyBenchmarks (Polo et al., 2024) and metabench (Kipnis et al., 2024), we directly compare our method with their released subsets[1][2].

Datasets. To comprehensively evaluate the applicability, following (Polo et al., 2024; Kipnis et al., 2024; Ying et al., 2024b), we selected four benchmarks covering diverse domains: GSM8K (Cobbe et al., 2021) for math reasoning; ARC-Challenge (Clark et al., 2018) for scientific reasoning; Hellaswag (Zellers et al., 2019) for commonsense inference and MMLU (Hendrycks et al., 2020) for general tasks. The detailed statistics result is shown in the Appendix.

Comparison Models. To thoroughly evaluate our method, we select 17 models of varying scales from diverse sources, including both open-source and closed-source models. The selection covers both models with significant capability gaps and those with similar performance to simulate real-world scenarios. We assess the data sampling method by comparing the relative performance of these models on the original dataset versus the sampled subsets. The evaluated models include: (1) Qwen series (Bai et al., 2023a; Team, 2024; Yang et al., 2024; Team, 2025); (2) LLaMA series (Chiang et al., 2023; Touvron et al., 2023; Grattafiori et al., 2024; Guo et al., 2025); (3) Gemma series (Team et al., 2024); (4) proprietary models (OpenAI, 2024; DeepMind, 2025). See Appendix for detailed model list and generation settings.

Metrics. To evaluate how well the selected subsets represent the full datasets, we consider several metrics to quantify the discrepancy between the subset and the original data. Specifically, we report **Spearman's correlation** (r_S) and **Kendall's** τ (r_K) between model performances derived from the selected subset and those from the full dataset (i.e. relative ranking), following previous work (Vivek et al., 2023; Polo et al., 2024). In addition, we measure prediction fidelity using the **Mean Absolute Error (MAE)** between model performance on the subset and on the full dataset (i.e. absolute performance). We use the correlation coefficient as the primary metric, as discussed in the Introduction. Following (Polo et al., 2024), for methods involving randomness (Random, K-Means and EffiEval), we repeat them $t = 5$ times and average the above metrics results to ensure robustness.

Implementation details For the threshold η of MUI in Eq. (3), we adopt a layer-wise top-k% threshold following (Cao et al., 2025b). See Appendix B for details. For the selection of k in Algorithm 1, we set the subset size to match the sizes of publicly available representative datasets to ensure fair comparison with baselines. Since existing representative subsets may not fully cover the evaluation capabilities of the full benchmark , we also

[1] https://huggingface.co/tinyBenchmarks
[2] https://huggingface.co/datasets/HCAI/metabench

consider an adaptive coverage-based k setting: that is, we select the smallest data size that achieves a predefined coverage threshold of $r = 0.8$, formally:

$$k^* = \min_k\{k, |N_{\text{covered}}(k)|/|N_{\text{covered}}(K)| \geq r\} \quad (7)$$

Unless specified, we choose LLaMA3.1-8B-Instruct as the indicator model. For more experiments using more different models, please refer to Section 4.5.

4.2 Effectiveness of EffiEval

For a fair comparison, we follow the settings of SOTA methods and sample the same number of evaluation samples from each original benchmark as they do. Note that this number may not be optimal for our method, and we will discuss the issue of subset size in later sections. The correlation results across the 17 selected models shown in Table 1 and Table 2 indicate that: 1) Our method achieves strong relative correlation across all datasets, with low correlation variance (please refer to the Appendix for details), demonstrating the effectiveness of our approach; 2) In the ARC and MMLU datasets shown in Table 2, our selection process exhibits slight sensitivity to the subset size and does not achieve optimal results. This may be because, when the number of samples is relatively small in these challenging datasets, the model's capabilities cannot be consistently and fully covered, leading to certain fluctuations. 3) On the MAE metric, the results vary significantly across different methods, suggesting that it may not be a reliable indicator — consistent with our earlier discussion on relative versus absolute performance scores; 4) Compared to r_K, the r_S metric is generally less sensitive to pairwise ranking errors, leading to consistently higher scores across different methods. In contrast, r_K provides a more fine-grained characterization of how well the sampling method preserves model rankings, and its values are generally lower. For example, on certain datasets such as GSM8K, even SOTA methods achieve only 89.3%. 5) The K-Means baseline does not show a significant advantage over the random baseline, yet it incurs higher computational overhead. This demonstrates that the random strategy is already a strong baseline. However, the effectiveness of the random method may be overestimated, as repeated sampling effectively increases the actual size of the subset. For the variability comparison, please refer to Appendix E.

4.3 Capability Coverage

SOTA methods typically select fixed, and small benchmark subsets (e.g., tinyBenchmarks with only 100 samples). We argue that the fixed size of these subsets makes it difficult to dynamically adjust their size, and that their selection is often insufficient to capture the full dataset's content coverage. In our in-depth analysis, we quantitatively examine the representativeness issues arising from such limited data. For example, as shown in Figure 2, the 100 representative samples selected by tinyBenchmarks and metabench cover only 46 and 37 out of 57 sub-tasks in the MMLU benchmark, respectively. From the perspective of model capability activation, the MUI activated by these subsets accounts for approximately 10% of that activated by the full dataset for the indicator model (see Table 12 in the Appendix for more details). These observations suggest that **such SOTA methods fail to select a representative subset with both appropriate size and distribution to ensure adequate coverage of model capabilities**. In contrast, selecting data based on MUI capability coverage can readily address the insufficient capability coverage of existing methods. Instead of searching for an appropriate subset size k for each target benchmark, we iteratively add data samples to the subset S using EffiEval until the coverage ratio reaches a threshold r. For example, when $r = 0.8$ is set to ensure high data quality, our method achieves complete category coverage on the MMLU dataset, covering all 57 subtasks. Importantly, this process allows the threshold to be flexibly adjusted based on efficiency requirements. Furthermore, from the perspective of effectively substituting for the original dataset, the results in Table 3 show that subsets selected by EffiEval demonstrate clear advantages over both random selection and K-means clustering approaches, yielding superior performance. **Overall, these results highlight the effectiveness and scalability of our approach.**

4.4 Determine Subset Size Adaptively

By selecting data to achieve a coverage ratio of $r = 0.8$, we have already demonstrated that the resulting subsets can effectively serve as substitutes for the full dataset. In this section, we further investigate the relationship between different coverage thresholds and the effectiveness of the selected subsets. To investigate this, we evaluate the effectiveness of representative subsets selected under

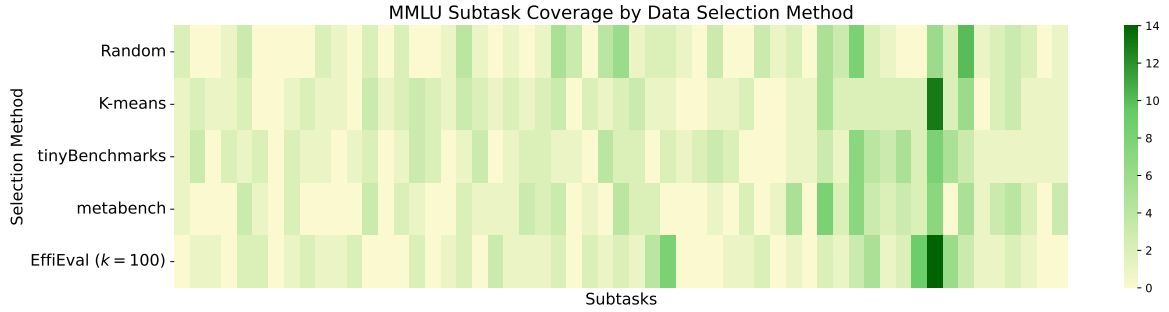

Figure 2: Category coverage of the representative dataset baselines and **EffiEval** (with sample number set to fixed $k = 100$) on MMLU (57 categories in total). Each cell's color intensity reflects how many samples are selected in the corresponding category. The heatmap illustrates that previous methods do not fully cover the categories of the dataset, indicating that the subset size k of these baselines may be improper. The horizontal axis corresponds to the 57 category labels of the MMLU dataset, while the vertical axis lists the different selection methods.

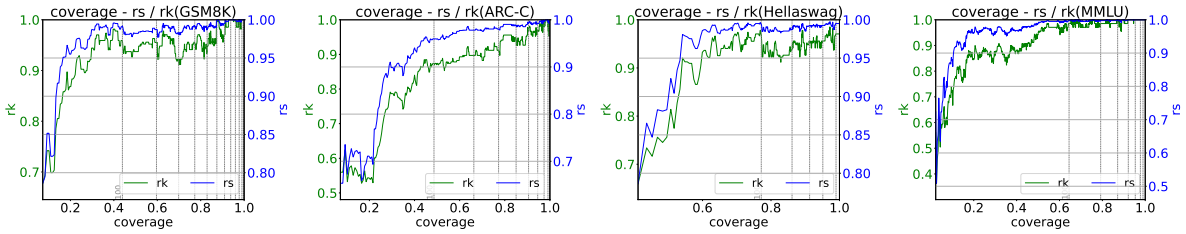

Figure 3: A dual-axis plot of r_K (green) and r_S (blue) versus coverage on four benchmarks. Gray vertical dashed lines mark the coverage levels corresponding to evenly spaced values of sample size k, with the step size indicated at the first line. The figure illustrates how much data is needed for different datasets to achieve a trade-off between correlation and evaluation efficiency.

Method	Dataset	k (k/K)	r_S	r_K	MAE ↓
Random	GSM8K	446 (33.8%)	98.9	94.6	**1.03**
K-means			99.2	96.1	1.40
EffiEval			**99.3**	**96.3**	1.99
Random	ARC	332 (28.3%)	98.6	93.8	**1.47**
K-means			98.7	94.3	1.52
EffiEval			**99.0**	**95.6**	1.48
Random	Hellaswag	125 (1.2%)	97.9	91.1	3.18
K-means			98.2	92.0	3.11
EffiEval			**98.5**†	**92.3**†	**2.78**†
Random	MMLU	2082 (14.8%)	99.4	96.8	**0.94**
K-means			99.7	98.0	1.97
EffiEval			**99.8**	**98.5**	**0.94**

Table 3: Effectiveness of the representative subset based on a coverage ratio $r = 0.8$. The efficiency is quantified by the ratio of the selected subset size to the full dataset size (k/K).

different coverage thresholds, following the experimental settings in Experiment Setting Section. The result shown in Figure 3 indicates that when r is small, the selected subset fails to sufficiently represent the full dataset, leading to low correlations. As r increases, the correlation improves, finally converges to 1. Moreover, Figure 3 reveals large variation in information density across benchmarks. For example, on Hellaswag, selecting just 125 samples (1% of the original dataset) covers over 80% of the neurons activated by the full dataset, whereas MMLU requires approximately 2000 samples (14.8% of the original dataset) to reach the same coverage. Notably, on certain benchmarks (e.g., MMLU), high correlations can be achieved even at lower coverage thresholds (e.g., $r = 0.6$). We hypothesize that this is because some capabilities — though controlled by different neurons — are highly interrelated. As a result, **it is possible to adaptively select an efficient coverage threshold based on the characteristics of different datasets, while still maintaining high evaluation effectiveness. This highlights the flexibility and adaptability of our approach.** To select a value of k that balances evaluation efficiency and reliability, we search across all possible k values for a window in which the correlation remains both high and stable. Specifically, we identify the earliest window of 200 consecutive k values that satisfies the following two conditions: 1) Reliability: Each k in the window yields a Kendall's τ correlation above 0.9; 2) Stability: The correlation values within the window are stable, meaning their Kendall's τ[3] with respect

[3]Note that this Kendall's τ measures the stability of the correlation sequence, and is different from the one used as our primary evaluation metric.

to a monotonically increasing index is less than 0.1 in absolute value. The midpoint of this window is then selected as the final value of k. Table 4 summarizes the evaluation metrics under this setting of k values. Compared with fixed-r selection, this strategy further reduces the subset size—by up to 95% — without sacrificing too much correlation.

Method	Dataset	k (k/K)	r_S	r_K	MAE ↓
Random	GSM8K	140 (10.6%)	96.3	89.5	**2.37**
K-means			96.6	89.7	2.44
EffiEval			**98.2**	**93.7**	4.38
Random	ARC	409 (34.9%)	98.6	93.8	1.33
K-means			**99.3**	**96.6**	**1.08**
EffiEval			98.7	94.1	1.60
Random	Hellaswag	149 (1.5%)	98.2	91.9	**2.72**
K-means			98.2	92.2	2.76
EffiEval			**98.5**†	**92.6**†	3.32†
Random	MMLU	669 (4.8%)	98.8	95.1	1.88
K-means			99.1	95.3	1.63
EffiEval			**99.5**	**97.1**	**1.55**

Table 4: Effectiveness of the representative subset based on stable sliding window.

4.5 Ablation Study

Independence between EffiEval and performance. A good selection process should be orthogonal to the models' performance, otherwise it could introduce bias. To verify that EffiEval is performance-independent, we repeatedly sample 10% of the smaller group between the correct and incorrect samples from four benchmarks, and repeat this process 10 times, then compute MUI of the indicator model on the two sets, finally conduct Mann-Whitney U test to verify that there is no statistically significant difference between the two distributions. As shown in Figure 4, no significant difference is observed across performance-based groupings, demonstrating that MUI is independent of model performance. This property guarantees that EffiEval selects representative samples in a performance-agnostic manner.

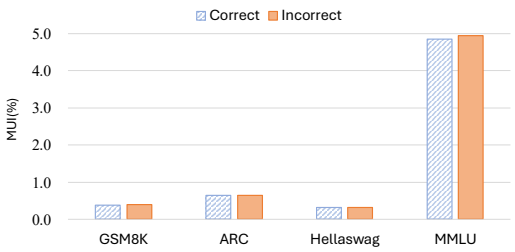

Figure 4: MUI of correct samples and incorrect samples on four benchmarks. The Mann-Whitney U test fails to reject the null hypothesis, suggesting that there is no significant difference between the two distributions.

Switching the indicator model. In the experiments above, we mainly use LLaMA-3.1-8B-Instruct to identify key neurons, suggesting that a single model is sufficient to select a representative subset that covers the capabilities of other models. We hypothesize that this may be due to a high degree of correlation in capability distributions across different models. To further investigate this, we explore how much the choice of model affects the selection outcome. For this purpose, we conduct additional experiments on selected benchmarks with Qwen2.5-7B-Instruct and Qwen-1.5-7B-Chat, with k set identical to those in Table 3. The experiment results are shown in Table 5. In most cases, replacing the indicator model still successfully guides the data selection process, achieving strong performance correlations. For the Qwen series on the MMLU benchmark, some of the selected questions are relatively simple, making it difficult to distinguish between models and resulting in low correlation scores. For further discussion of this phenomenon, see Section 6.

Method	**GSM8K** (k = 446)			**MMLU** (k = 2082)		
	r_S	r_K	MAE ↓	r_S	r_K	MAE ↓
Random	98.9	94.6	1.03	99.4	96.8	0.94
K-Means	99.2	96.1	1.40	99.7	98.0	1.97
LLaMA3.1-8B	99.3	96.3	1.99	99.8	98.5	0.94
Qwen2.5-7B	98.9	94.7	2.32	96.9	88.8	10.85
Qwen1.5-7B	99.1	95.2	1.09	97.9	91.2	9.41

Table 5: Comparison between different indicator models. The selection remains effective when switching the indicator model, showing strong performance correlations and supporting the generalization of our method. Results on other benchmarks are provided in Table 15 in the Appendix.

5 Conclusion

In this paper, we address the urgent need for efficient and reliable evaluation in the era of LLMs. We propose EffiEval, a training-free approach for benchmark subset selection that maximizes internal capability coverage as measured by the MUI. Our method is specifically designed to satisfy three key criteria for high-quality evaluation: representativeness, fairness, and generalizability. Extensive experiments on multiple public benchmarks demonstrate that EffiEval consistently achieves strong correlation with full-dataset evaluation. Moreover, our approach is flexible and scalable, enabling users to adjust the trade-off between evaluation efficiency and coverage.

6 Limitations

Despite the effectiveness of our method, several limitations remain and deserve further exploration. **1) Regarding data distribution**. Data distribution is vital for evaluation. Different scenarios or different stages of model training (e.g., evaluating pretrained vs. post-trained models) may require evaluation sets with distinct distributions (e.g., more diverse or more challenging and discriminative questions). In this paper, we prioritize diversity from the perspective of capability coverage, thus favoring a more diverse distribution. Under this setting, when the original data distribution contains redundant assessments of certain capabilities, our sampling method alters the original evaluation set's distribution, leading to a significant drop in correlation metrics. Although controlling the difficulty distribution can be achieved by incorporating penalty terms into the sampling method, future work should explore more reasonable and effective evaluation strategies that explicitly account for redundancy, diversity, and the actual capabilities being tested. **2) Regarding the selection of the indicator model**. In Section 4.5, we observe that the MUI produced by the Qwen models on the MMLU dataset exhibit minor differences between correct and incorrect samples. This results in selected subsets that are overly easy, making it difficult to distinguish between different models and consequently reducing correlation metrics. We suggest selecting an indicator model whose MUI scores are more evenly distributed across correct and incorrect samples. Additionally, we plan to explore methods to mitigate this bias in future work.

References

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, and 29 others. 2023a. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Yushi Bai, Jiahao Ying, Yixin Cao, Xin Lv, Yuze He, Xiaozhi Wang, Jifan Yu, Kaisheng Zeng, Yijia Xiao, Haozhe Lyu, Jiayin Zhang, Juanzi Li, and Lei Hou. 2023b. Benchmarking foundation models with language-model-as-an-examiner. In *Advances in Neural Information Processing Systems*, volume 36, pages 78142–78167. Curran Associates, Inc.

Yixin Cao, Shibo Hong, Xinze Li, Jiahao Ying, Yubo Ma, Haiyuan Liang, Yantao Liu, Zijun Yao, Xiaozhi Wang, Dan Huang, Wenxuan Zhang, Lifu Huang, Muhao Chen, Lei Hou, Qianru Sun, Xingjun Ma, Zuxuan Wu, Min-Yen Kan, David Lo, and 8 others. 2025a. Toward generalizable evaluation in the llm era: A survey beyond benchmarks. *Preprint*, arXiv:2504.18838.

Yixin Cao, Jiahao Ying, Yaoning Wang, Xipeng Qiu, Xuanjing Huang, and Yugang Jiang. 2025b. Model utility law: Evaluating llms beyond performance through mechanism interpretable metric. *arXiv preprint arXiv:2504.07440*.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Google DeepMind. 2025. Gemini 2.5 model family expands: Pro, flash, and flash-lite. https://blog.google/products/gemini/gemini-2-5-model-family-expands/. Published June 17, 2025.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

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*.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Alex Kipnis, Konstantinos Voudouris, Luca M Schulze Buschoff, and Eric Schulz. 2024. metabench – A Sparse Benchmark to Measure General Ability in Large Language Models. *arXiv preprint arXiv:2407.12844*.

Tianle Li, Wei-Lin Chiang, Evan Frick, Lisa Dunlap, Tianhao Wu, Banghua Zhu, Joseph E Gonzalez, and Ion Stoica. 2024. From crowdsourced data to high-quality benchmarks: Arena-hard and benchbuilder pipeline. *arXiv preprint arXiv:2406.11939*.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, and 1 others. 2022. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*.

George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. 1978. An analysis of approximations for maximizing submodular set functions—i. *Mathematical programming*, 14:265–294.

OpenAI. 2024. Gpt-4o: Openai's new flagship model.

Lorenzo Pacchiardi, Lucy G Cheke, and José Hernández-Orallo. 2024. 100 instances is all you need: predicting the success of a new llm on unseen data by testing on a few instances. *arXiv preprint arXiv:2409.03563*.

Haowen Pan, Yixin Cao, Xiaozhi Wang, Xun Yang, and Meng Wang. 2024. Finding and editing multi-modal neurons in pre-trained transformers. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 1012–1037, Bangkok, Thailand. Association for Computational Linguistics.

Yotam Perlitz, Elron Bandel, Ariel Gera, Ofir Arviv, Liat Ein-Dor, Eyal Shnarch, Noam Slonim, Michal Shmueli-Scheuer, and Leshem Choshen. 2023. Efficient benchmarking of language models. *arXiv preprint arXiv:2308.11696*.

Felipe Maia Polo, Lucas Weber, Leshem Choshen, Yuekai Sun, Gongjun Xu, and Mikhail Yurochkin. 2024. tinybenchmarks: evaluating llms with fewer examples. *arXiv preprint arXiv:2402.14992*.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adri Garriga-Alonso, and 1 others. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on machine learning research*.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, and 179 others. 2024. Gemma 2: Improving open language models at a practical size. *Preprint*, arXiv:2408.00118.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Qwen Team. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L Turner, Callum McDougall, Monte MacDiarmid, C. Daniel Freeman, Theodore R. Sumers, Edward Rees, Joshua Batson, Adam Jermyn, and 3 others. 2024. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. *Transformer Circuits Thread*.

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*.

Rajan Vivek, Kawin Ethayarajh, Diyi Yang, and Douwe Kiela. 2023. Anchor points: Benchmarking models with much fewer examples. *arXiv preprint arXiv:2309.08638*.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. 2024. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*.

Jiahao Ying, Yixin Cao, Yushi Bai, Qianru Sun, Bo Wang, Wei Tang, Zhaojun Ding, Yizhe Yang, Xuanjing Huang, and Shuicheng Yan. 2024a. Automating dataset updates towards reliable and timely evaluation of large language models. In *Advances in Neural Information Processing Systems*, volume 37, pages 17106–17132. Curran Associates, Inc.

Jiahao Ying, Mingbao Lin, Yixin Cao, Wei Tang, Bo Wang, Qianru Sun, Xuanjing Huang, and Shuicheng Yan. 2024b. LLMs-as-instructors: Learning from errors toward automating model improvement. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 11185–11208, Miami, Florida, USA. Association for Computational Linguistics.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*.

A Generation Settings

In all experiments, we adopt consistent decoding settings to ensure fair comparison across models. Specifically, we set the maximum number of newly generated tokens (`max_new_tokens`) to 8192 for reasoning models (DeepSeek-R1-Distill-Qwen-7B, DeepSeek-R1-Distill-LLaMA-8B and Qwen-3-8B with thinking enabled), and 1024 for all other models. We use top-p sampling with `top_p` = 1.0, and fix the temperature to 0.0 to disable sampling and enforce deterministic generation.

B Implementation for Threshold Function

In this paper, we adopt a layer-wise top-k% threshold function for key neuron selection, i.e.

$$\eta(l, topk) = V_l^{\lfloor N \cdot topk \rfloor}$$

Where V_l is the sorted (in descending order) activation scores on all neurons and in output tokens in layer l, $topk \in (0, 1)$ is the threshold hyperparameter, N is the neuron number per layer. In all experimental settings, we set $topk = 0.1\%$. This approach can alleviate discrepancies due to different model architectures when calculating the MUI (Cao et al., 2025b).

C Computation Cost Analysis

EffiEval can be divided into two stages: (1) neuron score computation and (2) solving the Maximum Coverage Problem. For Stage 1, according to Eq. (2), based on the forward pass, computing the neuron scores only requires two additional matrix multiplications (note that the term $W_{in}^l(\mathbf{x}_{-1}^l)$ is computed along with the forward pass). This overhead is negligible compared with the cost of the forward pass. On our computing hardware (see Appendix G), these extra computations take approximately 2 seconds for a batch of size 4 with sequence length 1024. For Stage 2, the time complexity of Algorithm 1 is $O(k \cdot KN)$, which is proportional to the subset size k. As an example, we list in Table 6 the computational time overhead of the subset selection process outlined in Table 3. Overall, our method does not introduce substantial computational cost.

D Case Study

In this section, we illustrate the execution process of EffiEval to better demonstrate how the algorithm selects samples with previously uncovered capabilities. For each candidate sample, we present its most similar counterpart (i.e., a sample whose capabilities have already been covered) and its most dissimilar one (i.e., the sample most likely to be selected by EffiEval in the next step). The similarity metric is defined as the Jaccard distance $J(\cdot, \cdot)$ between the sets of activated neurons:

$$\text{dist}(t_i, t_j) = J(N_{\text{activated}}(t_i), N_{\text{activated}}(t_j))$$

	GSM8K	ARC	Hellaswag	MMLU
Subset size k	446	332	125	2082
Cost (s)	7.4	5.5	21.8	124.8

Table 6: The computational overhead (seconds) of Stage 2 in Table 3. Our method does not introduce significant computational overhead.

Case Study on GSM8K

Prompt:
Digimon had its 20th anniversary. When it came out John was twice as old as Jim. If John is 28 now how old is Jim?
(One-variable linear equation)
Most similar prompt:
Liam is 16 years old now. Two years ago, Liam's age was twice the age of Vince. How old is Vince now?
(One-variable linear equation)
Most dissimilar prompt:
My kitchen floor has a total area of 200 SqFt. I want to install new square floor tiles that cost $12 each, and each tile side is 1ft in length. How much will it cost me to renovate my kitchen floor?
(Area Computation)

Case Study on ARC-Challenge

Prompt:
A student pushed a large rubber ball on a flat, frictionless surface. The ball rolled at a speed of 1 meter per second. Which statement best describes the motion of the ball when the student stopped pushing the ball?
A. The ball accelerated.
B. The ball did not move.
C. The ball changed direction.

D. The ball continued to move in the same direction.
(Physics)
Most similar prompt:
Two girls are pulling on opposite ends of a thick rope. Both girls pull on the rope with the same force but in opposite directions. If both girls continue to pull with the same force, what will most likely happen?
A. One girl will pull the other toward her.
B. Both girls will stay in the same place.
C. Gravity will cause the rope to sag.
D. The rope will break.
(Physics)
Most dissimilar prompt:
Which question about tulips could best be answered by scientific research?
A. Are tulips better than other flowers?
B. What genes determine tulip petal color?
C. Why do people like to look at tulips?
D. Which color of tulips is the prettiest?
(Biology)

Case Study on Hellaswag

Prompt:
[header] How to use and install a live cd of linux [title] Make sure your computer is booting from the cd drive. [substeps] Either turn on or restart your computer. While doing this, hold the delete button to enter the bios.
1. [title] Once you've backed up your computer, press a. [step] After a few seconds the bios should pop up on your screen.
2. If you don't hear any result after a few seconds of working (or if the cd drive isn't booting freely from the computer), you may need to reboot your computer. [title] Run the cd containing the bios.
3. Use your left and right arrow keys to navigate to the boot tab. * once on the boot tab use your down arrow keys to navigate to the "boot device priority" menu.
4. The program is now on the cd drive and it cannot be added to the system at any time. To quit the program via the bios), enter the status key and select restart the computer.
(Operating System)
Most similar prompt:
[header] How to reformat windows 7 [title] Backup all your files, drivers and settings so that you can restore them later. [title] Find all your installation discs or product keys for the programs you want to keep so that you can restore them after the installation is complete. [title] Partition your hard drive.
1. [step] For windows 7 you will need to partition all of your data, but this is optional as your computer needs your hard drive. [substeps] Right-click your drive and select partition all.
2. [step] Partition your hard drive to remove your usb storage device from cd or dvd. [title] Finish the process using back up documentation if you wish to keep a backup device.
3. [step] This means dividing the hard drive into parts and making the parts available to the os (operating system). [title] Click on "start" and then control panel.
4. [step] This helps to prevent any lost drives. You can also patch any issues that you have with windows 7.
(Operating System)
Most dissimilar prompt:
A woman is bent over holding a weight bar. She picks the weight up and holds it at her shoulders. she
1. then pushes off the bar.
2. bends down and begins exercising using the weight bar.
3. then lifts the weight over her head.
4. lifts it to her chest and works out.
(Commonsense Understanding)

Case Study on MMLU

Prompt:
In the absence of intervention, imperfect competition, externalities, public goods, and imperfect information all result in which of the following?
A. Demand curves that should be added vertically
B. Market failure
C. Prices that are too low
D. Quantities of output that are too high
(High School Microeconomics)
Most similar prompt:

A negative externality in the market for a good exists when
A. the market overallocates resources to the production of this good.
B. spillover benefits are received by society.
C. the marginal social benefit equals the marginal social cost.
D. total welfare is maximized.
(High School Microeconomics)
Most dissimilar prompt:
In the current year Vinton exchanged unimproved land for an apartment building. The land had a basis of $300000 and a fair market value (FMV) of $420000 and was encumbered by a $100000 mortgage. The apartment building had an FMV of $550000 and was encumbered by a $230000 mortgage. Each party assumed the other's mortgage. What is Vinton's basis in the office building?
A. $300,000
B. $320,000
C. $430,000
D. $550,000
(Professional Accounting)

E Variability Analysis

In this section, we verify the variability of the selection process using the Generalized Jaccard Index, which is defined as

$$J(S_1, S_2, ..., S_t) = \frac{|\bigcap_{i=1}^{t} S_i|}{|\bigcup_{i=1}^{t} S_i|}$$

where S_i denotes the sample index set obtained from the i-th run of the random selection process. A lower value of this metric indicates that the selected subsets vary more across different runs, suggesting that the selection process is less stable. Conversely, a higher value implies greater stability and consistency in the selected samples. We fix $k = 100$ and repeat the random selection process $t = 5$ times. The results are shown in Table 7. As indicated by the Generalized Jaccard Index, our method consistently selects highly overlapping subsets, even when the subset size is small, demonstrating a high degree of stability.

F Separability Analysis

In this section, we assess the effectiveness of various data sampling strategies in distinguishing model performance using the Separability with Confidence metric (Li et al., 2024). This metric quantifies how confidently a benchmark can separate different models by computing the proportion of model pairs whose performance intervals do not overlap under repeated evaluations. Specifically, we apply each sampling method multiple times to generate subsets of evaluation data, and measure how often the subset can reliably identify a performance difference between two models. To obtain reliable percentile estimates for the confidence intervals, we set the number of sampling repetitions to $t = 50$. A higher separability score indicates that the sampling method tends to produce subsets that preserve meaningful model distinctions, providing stronger signals for model comparison. As shown in Table 8, our method achieves a higher separability score compared to random baselines, indicating a more stable and confident distinction between model performances. This suggests that our selected subsets tend to yield more consistent model rankings across repeated evaluations.

Method	GSM8K	ARC	Hellaswag	MMLU
Random	0.0	0.0	0.0	0.0
K-Means	1.1	1.5	**6.1**	8.0
EffiEval	**60.5**	**79.6**	5.9	**44.5**

Table 7: Generalized Jaccard Index of different selection methods across $t = 5$ runs on four benchmarks ($k = 100$). A higher Jaccard Index reflects greater stability in the selected subsets.

Method	GSM8K	ARC	Hellaswag	MMLU
Random	55.2	52.2	68.4	44.9
K-Means	60.3	55.2	**77.9**	55.2
EffiEval	**83.8**	**89.0**	73.5	**82.4**

Table 8: Separability with Confidence scores of different sampling methods across four benchmarks ($k = 100$, confidence level=95%). Our method outperforms the random baseline on all benchmarks and achieves significantly higher separability than K-Means on three out of four benchmarks, with one benchmark showing slightly lower performance than K-Means. Overall, this demonstrates the strong and stable capability of our method to distinguish model performances across diverse tasks.

G Computing Infrastructure

All experiments were conducted on a server running Ubuntu 22.04.5 LTS, equipped with an In-

tel(R) Xeon(R) Platinum 8480+ CPU, 2TB of RAM, and a 96GB NVIDIA H20 GPU.

H Use Of AI Assistants

During the preparation of this manuscript, we used large language models (LLMs) solely for assistance with grammar correction and phrasing improvements. All scientific content, experimental analyses, methodological decisions, and citations were conceived, critically reviewed, and validated entirely by the authors, who retain full responsibility for the integrity and accuracy of the work.

Method	**GSM8K** (k = 100)		**ARC** (k = 100)		**Hellaswag** (k = 100)		**MMLU** (k = 100)	
	std.$(r_S)\downarrow$	std.$(r_K)\downarrow$	std.$(r_S)\downarrow$	std.$(r_K)\downarrow$	std.$(r_S)\downarrow$	std.$(r_K)\downarrow$	std.$(r_S)\downarrow$	std.$(r_K)\downarrow$
Random	2.70	4.55	2.49	4.78	1.11	3.61	2.20	4.82
K-Means	2.99	4.92	1.73	3.76	0.65	2.37	1.63	3.65
EffiEval	**0.33**	**1.15**	**0.66**	**1.03**	**0.37**†	**0.84**†	**0.44**	**1.66**

Table 9: Standard deviation(std.) of correlation(r_S, r_K) in Table 1. Compared with random baselines, our method is more stable to select a representative subset.

Method	**GSM8K** (k = 446)		**ARC** (k = 332)		**Hellaswag** (k = 125)		**MMLU** (k = 2082)	
	std.$(r_S)\downarrow$	std.$(r_K)\downarrow$	std.$(r_S)\downarrow$	std.$(r_K)\downarrow$	std.$(r_S)\downarrow$	std.$(r_K)\downarrow$	std.$(r_S)\downarrow$	std.$(r_K)\downarrow$
Random	0.86	2.55	0.72	2.34	0.99	3.69	0.33	1.44
K-Means	0.41	1.27	0.61	1.91	0.82	3.72	0.36	1.76
Llama3.1	0.25	1.16	0.27	1.24	0.50	1.46	0.18	1.10
Qwen2.5	0.20	0.68	0.18	**0.62**	**0.37**	**1.44**	0.45	**0.72**
Qwen1.5	**0.19**	**0.57**	**0.13**	0.91	0.42	1.57	**0.11**	0.99

Table 10: Standard deviation(std.) of correlation(r_S, r_K) in Table 3 and Table 5. Compared with random baselines, our method is more stable to select a representative subset.

Method	**GSM8K** (k = 140)		**ARC** (k = 409)		**Hellaswag** (k = 149)		**MMLU** (k = 669)	
	std.$(r_S)\downarrow$	std.$(r_K)\downarrow$	std.$(r_S)\downarrow$	std.$(r_K)\downarrow$	std.$(r_S)\downarrow$	std.$(r_K)\downarrow$	std.$(r_S)\downarrow$	std.$(r_K)\downarrow$
Random	1.37	2.87	0.45	2.19	0.51	2.06	0.91	2.66
K-Means	1.07	3.01	0.63	2.32	0.52	1.53	0.72	2.83
EffiEval	**0.26**	**1.34**	**0.16**	**0.57**	**0.18**†	**0.90**†	**0.11**	**0.64**

Table 11: Standard deviation(std.) of correlation(r_S, r_K) in Table 4. Compared with random baselines, our method is more stable to select a representative subset.

Method	**GSM8K**	**ARC**	**Hellaswag**	**MMLU**
Random	30.1	31.6	60.8†	11.5
K-Means	30.3	29.9	60.5†	10.9
tinyBenchmarks	30.6	33.8	61.0†	11.0
metabench	46.0	39.5	60.7†	10.6

Table 12: Neuron coverage ratio r of the baselines in Table 1 and Table 2. Except for metabench, whose k is set to its originally selected value, all other entries use a fixed k of 100.

Model Series	Model Name
Qwen	Qwen-1.5-7B-Chat (Bai et al., 2023a) Qwen-2.5-1.5B-Instruct Qwen-2.5-7B-Instruct Qwen-2.5-14B-Instruct Qwen-2.5-32B-Instruct (Team, 2024) Qwen-2.5-Math-7B (Yang et al., 2024) Qwen-3-8B (thinking mode on/off) (Team, 2025) DeepSeek-R1-Distill-Qwen-7B
LLaMA	Vicuna-7B-v1.3 (Chiang et al., 2023) LLaMA-2-7B-Chat LLaMA-2-13B-Chat (Touvron et al., 2023) LLaMA-3.1-8B-Instruct (Grattafiori et al., 2024) DeepSeek-R1-Distill-LLaMA-8B (Guo et al., 2025)
Gemma	Gemma-2-9B-it (Team et al., 2024)
Proprietary	GPT-4o-2024-11-20 (OpenAI, 2024) Gemini-2.5-Flash-Preview-04-17 (DeepMind, 2025)

Table 13: List of evaluated models.

Statistic	GSM8K	ARC	Hellaswag	MMLU	Total
# samples	1319	1172	10042	14042	26575

Table 14: Statistics of datasets.

Method	**GSM8K** (k = 446)			**ARC** (k = 332)			**Hellaswag** (k = 125)			**MMLU** (k = 2082)		
	r_S	r_K	MAE ↓	r_S	r_K	MAE ↓	r_S	r_K	MAE ↓	r_S	r_K	MAE ↓
Random	98.9	94.6	1.03	98.6	93.8	1.47	97.9	91.1	3.18	99.4	96.8	0.94
K-Means	99.2	96.1	1.40	98.7	94.3	1.52	98.2	92.0	3.11	99.7	98.0	1.97
Llama3.1-8B	99.3	96.3	1.99	99.0	95.6	1.48	95.1^{-}	86.7^{-}	2.94^{-}	99.8	98.5	0.94
Qwen2.5-7B	98.9	94.7	2.32	98.8	94.5	2.76	98.5	92.3	2.78	96.9	88.8	10.85
Qwen1.5-7B	99.1	95.2	1.09	98.9	94.9	1.90	99.0	95.9	4.23	97.9	91.2	9.41

Table 15: Comparison between different indicator models on all four benchmarks. Due to safety constraints, LLaMA refuses to answer certain samples in Hellaswag (marked with $^{-}$), making it unable to effectively guide the data selection process.