Towards Optimal Evaluation Efficiency for Large Language Models

Anonymous ACL submission

Abstract

Comprehensive evaluation of large language models (LLMs) typically requires large-scale benchmarks, which is costly in terms of both data annotation and computational resource needed for evaluation. To mitigate these challenges, We propose an efficient evaluation framework that selects a question subset based on pre-tested results, thereby reducing the costs. We formulate the subset selection problem as an optimization task, solved using optimal random sampling and simulated annealing algorithms. We compare our approach with prior clustering-based methods and assess their reliability in terms of score accuracy. Additionally, we perform semantic analysis and evaluate whether the selected subsets preserve the semantic information of the original benchmark using Wasserstein distance. Experimental results show that our method outperforms previous approaches in terms of reliability, as measured by L2 norm. Our study provides an optimized perspective for balancing evaluation efficiency and reliability in LLM assessments, while revealing the relationship between optimization methods and semantic retention.

1 Introduction

Large language models (LLMs) demonstrate strong and generalizable capabilities. To evaluate these models comprehensively and accurately, large-scale benchmarks such as MMLU (Hendrycks et al., 2020), GSM8K (Cobbe et al., 2021), and HellaSwag (Zellers et al., 2019) are often required. However, these evaluations are expensive, consuming significant time, computational resources, or API tokens (Liang et al., 2022).

To alleviate these issues, efficient evaluation has recently gained growing attention. Vivek et al. (2023) have proposed clustering based on semantic information, using a subset of questions to predict the answers for the remaining ones. Polo et al. (2024) have noted that evaluation benchmarks usually have a large number of pre-tested model results, which can be explored to construct more reliable and efficient evaluation datasets.

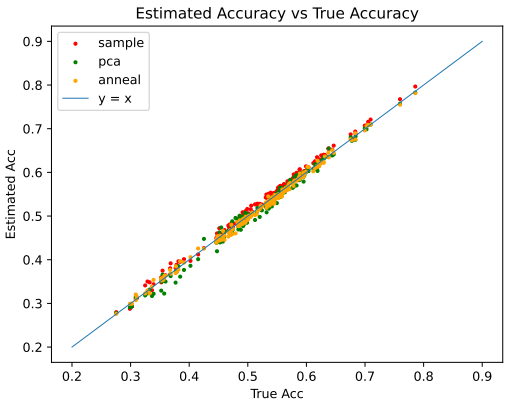

Figure 1: Comparison of estimated accuracy and true accuracy on MMLU for an evaluated LLM, using three methods to select a subset of only 19% of the original evaluation dataset: random sampling (sample), PCA-based clustering (pca), and simulated annealing (anneal). A more detailed quantitative comparison is provided in Section 4.

Nonetheless, these methods rely on heuristic selection based on clustering. We argue that instead of extracting relations between questions and performing selection through clustering, directly formulating an optimizable objective function can more clearly improve evaluation reliability, achieving better results. An empirical comparison is shown in Figure 1.

We hence propose an efficient evaluation where a combinatorial optimization perspective is explored. Specifically, the subset selection problem is reformulated as a combinatorial optimization task. We improve previous clustering methods and compare them with the proposed new approach. We assess the reliability of these methods in addressing model evaluation efficiency using the L2 norm, indicating

how well the selected subset predicts the full set's results. Additionally, we conduct semantic analysis to visualize the semantic distribution of these subsets and compute the Wasserstein distance to evaluate how well the subset selection strategies preserve the semantic information of the original evaluation dataset.

Experimental results demonstrate that our method outperforms previous methods in terms of reliability for specific compression rates, providing a new perspective for balancing efficiency and reliability in LLM evaluations, while also revealing the relationship between optimization methods and semantic preservation.

2 Problem Statement

Given a benchmark $\boldsymbol{B}$ with a question set $\boldsymbol{S} = \{q_1, q_2, \ldots, q_n\}$, and a set of m models $\boldsymbol{L} = \{l_1, l_2, \ldots, l_m\}$, each model l_j answers the i-th question q_i in $\boldsymbol{S}$, and is scored by an evaluator in $\boldsymbol{B}$, yielding a score $Y_{i,j} \in [0, 1]$. For questions with clear correct answers (e.g., multiple-choice), the score is binary, $Y_{i,j} \in \{0, 1\}$.

The accuracy of model l_j on $\boldsymbol{S}$ is the mean score:

$$A(\boldsymbol{S}, l_j) = \frac{1}{n} \sum_{i=1}^{n} Y_{i,j}.$$

And the overall evaluation of $\boldsymbol{L}$ on S is represented as a vector:

$$\boldsymbol{R}(\boldsymbol{S}, \boldsymbol{L}) = (A(\boldsymbol{S}, l_1), A(\boldsymbol{S}, l_2), \ldots, A(\boldsymbol{S}, l_m)).$$

Next, we aim to compress the question set. Given a compression rate c, we select a subset $\boldsymbol{S}' \subset \boldsymbol{S}$ such that $n' = \lfloor cn \rfloor$. Evaluating models on $\boldsymbol{S}'$, we get new scores $Y'_{i,j} \in [0, 1]$, and the accuracy of model l_j on $\boldsymbol{S}'$ is:

$$A(\boldsymbol{S}', l_j) = \frac{1}{n'} \sum_{i=1}^{n'} Y'_{i,j}.$$

The overall evaluation of $\boldsymbol{L}$ on $\boldsymbol{S}'$ is:

$$\boldsymbol{R}(\boldsymbol{S}', \boldsymbol{L}) = (A(\boldsymbol{S}', l_1), A(\boldsymbol{S}', l_2), \ldots, A(\boldsymbol{S}', l_m)).$$

To measure the reliability of $\boldsymbol{S}'$ in approximating $\boldsymbol{S}$, we define a loss function $Q(\boldsymbol{S}, \boldsymbol{S}', \boldsymbol{L})$, which measures the difference between the evaluation results of $\boldsymbol{S}$ and $\boldsymbol{S}'$, using L2 norm:

$$\begin{aligned} Q(\boldsymbol{S}, \boldsymbol{S}', \boldsymbol{L}) &= \|\boldsymbol{R}(\boldsymbol{S}, \boldsymbol{L}) - \boldsymbol{R}(\boldsymbol{S}', \boldsymbol{L})\|_2 \\ &= \sqrt{\sum_{j=1}^{m} (A(\boldsymbol{S}, l_j) - A(\boldsymbol{S}', l_j))^2} \\ &= \sqrt{\sum_{j=1}^{m} \left(\frac{\sum_{i=1}^{n} Y_{i,j}}{n} - \frac{\sum_{i=1}^{n'} Y'_{i,j}}{n'} \right)^2}. \end{aligned}$$

Our goal is to select the subset $\boldsymbol{S}'$ using a subset selection strategy such that the value of the above Q function is minimized, i.e., to maximize the reliability of predicting the full evaluation scores. It is important to note that during the selection of $\boldsymbol{S}'$, we do not have direct access to the LLM set $\boldsymbol{L}$, and therefore cannot directly optimize the Q function. This leads to the various strategies discussed in the following sections.

3 Methodology

Fortunately, benchmarks that require efficient evaluation typically have a large number of pre-tested model results. Let the benchmark $\boldsymbol{B}$ have a set of pre-tested models $\hat{\boldsymbol{L}}$, which contains $\hat{m}$ models. The score $\hat{Y}_{i,j} \in [0, 1]$ of model $\hat{l}_j \in \hat{\boldsymbol{L}}$ on question $q_i \in \boldsymbol{S}$ is known. We can use these existing results to reasonably select the subset $\boldsymbol{S}'$, which will be used to evaluate the set of models $\boldsymbol{L}$, thus constructing an efficient evaluation framework based on pre-tested model results, as shown in Figure 2. So how do we set up a reasonable subset selection strategy?

Saranathan et al. (2024) has shown that random sampling is already a good baseline method. In clustering approaches, we treat the scores of question q_i on all models $\hat{l}_j$ as the embedding of that question, i.e., $(\hat{Y}_{i,1}, \hat{Y}_{i,2}, \ldots, \hat{Y}_{i,m})$. Then, we apply K-Means (Hastie, 2009) clustering to form n' clusters and compute the size and center of each cluster. The visualized clustering results can be found in Section B of Appendix. We take the center of each cluster as the selected subset $\boldsymbol{S}'$, and assign a weight to it proportional to the cluster size.

Further, we observe that we can directly estimate the Q function using pre-tested results, and optimize this estimate. For any selected subset $\boldsymbol{S}'$, the scores $\hat{Y}'_{i,j} \in [0, 1]$ of the pre-tested models on $\boldsymbol{S}'$ are also known. Therefore, we obtain an estimate

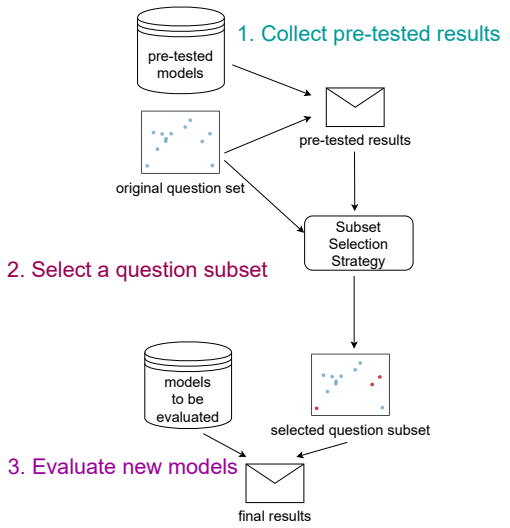

Figure 2: Efficient evaluation framework based on pre-tested results: Select a question subset based on existing results and use this subset to evaluate new models.

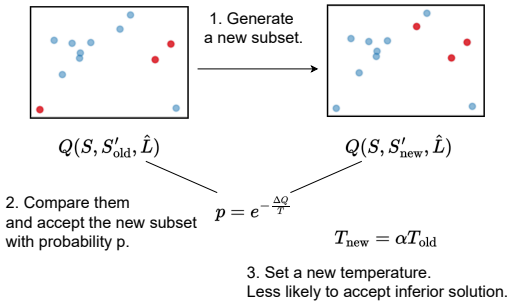

Figure 3: Simulated annealing process: In each iteration, a new solution is generated by replacing one element of the subset. The acceptance of the new solution is determined by a combination of solution quality and temperature. The process continues with iterative cooling until the target temperature is reached.

Dataset	#Questions	Evaluation Scope	Answer Format
MMLU	14,042	Comprehensive	Multiple-Choice
HellaSwag	10,042	Common Sense	Multiple-Choice
GSM8K	1,319	Mathematics	Deterministic

Table 1: Description of selected benchmarks.

of the Q function:

$$\hat{Q}(\boldsymbol{S}, \boldsymbol{S}', \boldsymbol{L}) = Q(\boldsymbol{S}, \boldsymbol{S}', \hat{\boldsymbol{L}}) = \sqrt{\sum_{j=1}^{\hat{m}} \left(\frac{\sum_{i=1}^{n} \hat{Y}_{i,j}}{n} - \frac{\sum_{i=1}^{n'} \hat{Y}'_{i,j}}{n'} \right)^2}$$

This is a typical combinatorial optimization problem with the constraint that the subset size is fixed. We can optimize the objective function using optimal random sampling or simulated annealing (Kirkpatrick et al., 1983). The optimal random sampling method performs multiple rounds of random sampling and selects the solution that minimizes the objective function as the final $\boldsymbol{S}'$. On the other hand, the simulated annealing method starts with a random initial solution and perturbs it in each iteration. It decides whether to accept the new solution based on the quality of the new solution and the current temperature. The higher the temperature, the greater the probability of accepting inferior solutions. Specifically, the probability is given by:

$$p = e^{-\frac{\Delta Q}{T}}$$

where T is the current temperature, which decreases exponentially after each round, i.e., $T_{\text{new}} = \alpha T_{\text{old}}, \alpha < 1$. And ΔQ represents the change in the Q function between the new and old solutions, i.e., $\Delta Q = Q(\boldsymbol{S}, \boldsymbol{S}'_{\text{new}}, \hat{\boldsymbol{L}}) - Q(\boldsymbol{S}, \boldsymbol{S}'_{\text{old}}, \hat{\boldsymbol{L}})$. If $\Delta Q < 0$, the new solution is always accepted. The iterative details of the subset selection using the simulated annealing method are shown in Figure 3. In this iteration process, we also record the solution that minimizes Q as the final $\boldsymbol{S}'$.

To assess semantic preservation, we embed all questions in the evaluation dataset and compute the Wasserstein distance between the subset and the full dataset, quantifying how well the subset retains the original dataset's semantic information. We solve the Wasserstein distance using the Sinkhorn algorithm (Cuturi, 2013), applying entropy regularization for faster computation.

4 Experiment

We selected open-source evaluation benchmarks with abundant pre-tested model results from the Open LLM Leaderboard (Beeching et al., 2023). For a given compression rate, we performed efficient evaluations using different subset selection strategies and analyzed their reliability and semantic preservation.

The benchmarks we selected include MMLU (Hendrycks et al., 2020), GSM8K (Cobbe et al., 2021), and HellaSwag (Zellers et al., 2019), each differing in scale, the range of evaluation capabilities, and the format of standard responses (Table 1), demonstrating the versatility of our method across different types of evaluation benchmarks.[1]

[1]These datasets and results are licensed for research use.

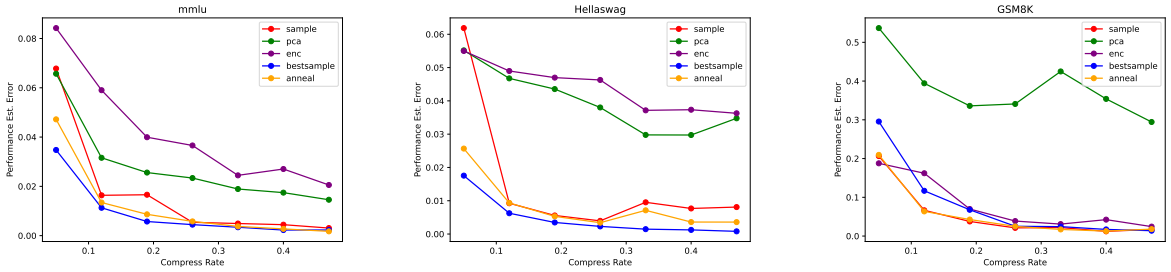

Figure 4: Comparison of evaluation results between the full benchmark $\boldsymbol{S}$ and the selected subsets $\boldsymbol{S}'$ across different compression rates, using five methods: random sampling (sample), PCA-based clustering (pca), autoencoder-based clustering (enc), optimal random sampling (bestsample), and simulated annealing (anneal).

The methods compared include random sampling, clustering-based methods using pre-tested model results, and combinatorial optimization methods. For the clustering-based method, we improved it by applying dimensionality reduction techniques. Polo et al. (2024) has shown that the performance of the same problem across different models is correlated, so the dimensions of the embedding have high information redundancy. We applied both PCA and autoencoder methods to reduce its dimensionality. For the combinatorial optimization methods, we used optimal random sampling and simulated annealing, both set with a maximum of 100 iterations.

For three benchmarks, we selected subsets $\boldsymbol{S}'$ with varying compression rates using the five methods and evaluated the model set $\boldsymbol{L}$. The results were compared with the full set $\boldsymbol{S}$, and the L2 norm of the difference vector was used as the error metric. The results are shown in Figure 4.

The experimental results indicate that:

1. Pure random sampling performs well on average, but its stability is poor due to the influence of the random seed, making further exploration of other methods necessary.

2. Random sampling and combinatorial optimization methods result in the smallest error.

3. Different dimensionality reduction techniques affect clustering results, which may be due to the intrinsic properties of the embedding. For example, for the GSM8K dataset, the pure linear mapping of PCA does not preserve the structure of the embedding well, while the autoencoder, which adds non-linear layers, can do so effectively.

To evaluate the semantic preservation, we conducted a semantic analysis on the MMLU evaluation benchmark. The methods compared include random sampling, clustering with PCA reduction, and combinatorial optimization with simulated annealing. We used Sentence-BERT (Reimers, 2019) for embedding and reduced the features to a two-dimensional space to visualize the semantic distribution, as shown in Section C of Appendix. The Wasserstein distances obtained at different compression rates are presented in Table 2.

	0.05	0.12	0.19	0.26	0.33	0.40
Sample	1.2149	**1.1069**	**1.0050**	**0.9127**	**0.8209**	**0.7341**
PCA	*1.2148*	1.1163	1.0327	0.9491	0.8675	0.7901
Anneal	**1.2105**	*1.1113*	*1.0090*	*0.9151*	*0.8244*	*0.7362*

Table 2: Wasserstein distances between subsets selected by different methods and the original set at different compression rates. The top row lists the corresponding compression rates c, with **bold** indicating the closest and *italics* indicating the second closest.

The results show that the preservation of semantic information correlates with evaluation accuracy. However, the random sampling method preserves semantic information better than the simulated annealing method, reflecting some of the costs associated with the single optimization goal.

5 Conclusion

In this paper, we have presented a combinatorial optimization approach for efficiently evaluating LLM capabilities. We introduce a novel evaluation framework and provide a comprehensive comparison, analyzing semantic retention to ensure the subset's alignment with the original benchmark.

Our work offers a new perspective on optimizing the balance between evaluation efficiency and reliability, highlighting key insights into the relationship between optimization techniques and semantic preservation.

Limitations

Through experiments, we have shown the correlation between the semantic retention of subsets and evaluation accuracy. However, this correlation is not absolute. For instance, the simulated annealing method slightly underperforms random sampling in terms of semantic retention, which reflects some limitations of our approach. For example, in tasks requiring high semantic fidelity, our approach may not be sufficiently applicable.

Future work may involve further refinement of the optimized function, including exploring different evaluation criteria and subset requirements, to investigate the generalizability of the combinatorial optimization approach.

Optimizing methods combined with semantic analysis may also be an interesting direction, exploring whether it is possible to optimize a given objective function while retaining semantic information, potentially further improving the robustness of subset for evaluating new models.

References

Edward Beeching, Clémentine Fourrier, Nathan Habib, Sheon Han, Nathan Lambert, Nazneen Rajani, Omar Sanseviero, Lewis Tunstall, and Thomas Wolf. 2023. Open llm leaderboard. *Hugging Face*.

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

Marco Cuturi. 2013. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26.

Trevor Hastie. 2009. The elements of statistical learning: data mining, inference, and prediction.

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*.

Scott Kirkpatrick, C Daniel Gelatt Jr, and Mario P Vecchi. 1983. Optimization by simulated annealing. *science*, 220(4598):671–680.

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

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*.

N Reimers. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Gayathri Saranathan, Mahammad Parwez Alam, James Lim, Suparna Bhattacharya, Soon Yee Wong, Martin Foltin, and Cong Xu. 2024. Dele: Data efficient llm evaluation. In *ICLR 2024 Workshop on Navigating and Addressing Data Problems for Foundation Models*.

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

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

A Autoencoder Settings

In the clustering method, we use a simple autoencoder to reduce 200-dimensional real-valued vectors to a 20-dimensional space. The encoder part consists of three fully connected layers followed by a ReLU activation function, while the decoder is symmetric to the encoder. We use the embeddings from the pre-tested results as training data to ensure that the autoencoder effectively preserves the structure of the data after dimensionality reduction.

B Cluster Results

We present visualizations of clustering results using PCA, derived from different scenarios in MMLU, with compress rate 0.05, to help better understand how clustering based on pre-tested model results can be applied to efficient LLM evaluation, as Figure 5, 6, 7.

C Semantic Distribution

We present the visualized results of question subsets obtained by different methods, after sentence-BERT embedding and PCA dimensionality reduction, to show their retention of the original evaluation dataset's semantics, as Figure 8, 9. These results may be helpful for future work on efficient LLM evaluation incorporating semantics.

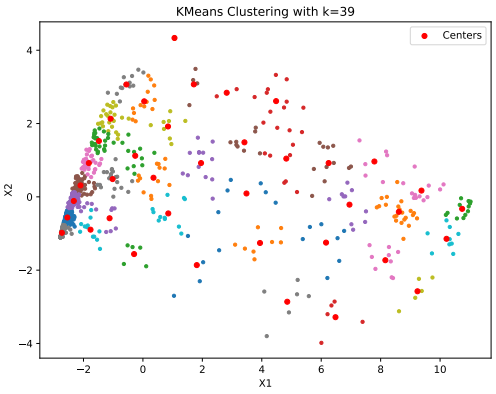

Figure 5: Visualization of clustering results for the scenario "miscellaneous" of MMLU, yielding a subset of size 39.

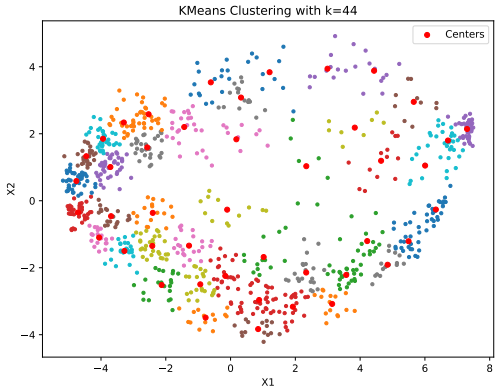

Figure 6: Visualization of clustering results for the scenario "moral_scenarios" of MMLU, yielding a subset of size 44.

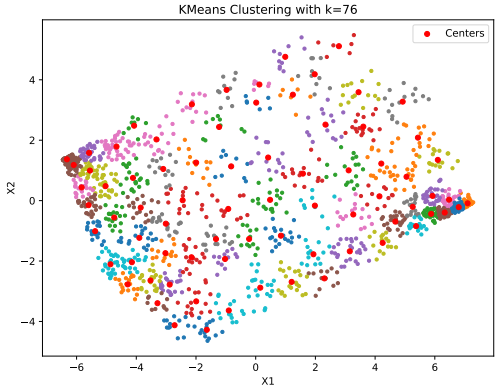

Figure 7: Visualization of clustering results for the scenario "professional_law" of MMLU, yielding a subset of size 76.

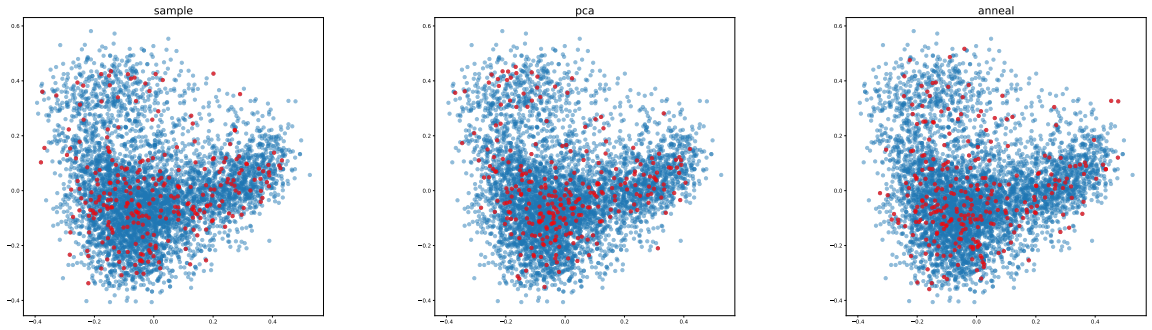

Figure 8: Semantic distribution visualization of MMLU at a 0.05 compression rate for different methods.

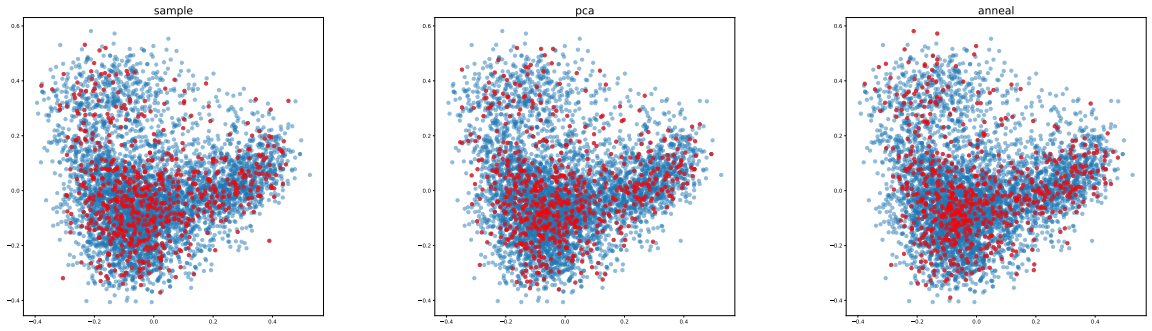

Figure 9: Semantic distribution visualization of MMLU at a 0.12 compression rate for different methods.