Poor-Supervised Evaluation for SuperLLM via Mutual Consistency

Peiwen Yuan[1], **Shaoxiong Feng**[2], **Yiwei Li**[1], **Xinglin Wang**[1], **Boyuan Pan**[2]
Heda Wang[2], **Yao Hu**[2], **Kan Li**[1*]
[1]School of Computer Science and Technology, Beijing Institute of Technology
[2]Xiaohongshu Inc
{peiwenyuan,liyiwei,wangxinglin,likan}@bit.edu.cn
{shaoxiongfeng2023,whd.thu}@gmail.com
{panboyuan,xiahou}@xiaohongshu.com

Abstract

The guidance from capability evaluations has greatly propelled the progress of human society and the development of Artificial Intelligence. However, as LLMs evolve, it becomes challenging to construct evaluation benchmark with accurate labels for LLMs whose capabilities approach or even surpass those of humans (denoted as **SuperLLMs**). To credibly conduct evaluation without accurate labels (denoted as **poor-supervised evaluation**), we first prove that the consistency between the model under evaluation and the reference model, when their prediction distributions are independent and the sample size is infinite, can equivalently assess the true capabilities of the model to be evaluated. However, using either humans or LLMs as the reference model cannot sufficiently meet the conditions, for which we propose the PEEM algorithm. By treating all models under evaluation as reference models, PEEM alternately optimizes model weights and filters reference models based on EM algorithm to maximally alleviate the insufficiency of the conditions. Comprehensive experiments across 3 types of tasks with 16 mainstream LLMs validate the efficiency, universality, and effectiveness of PEEM. More generally, PEEM has advanced the evaluation paradigm evolution from human-centric to human&model-centric, alleviating the limitations of human capabilities for evaluating SuperLLMs[1].

1 Introduction

Evaluations, such as IQ tests and Olympic competitions, effectively identify areas of strength and weakness for individuals or groups, providing valuable guidance for the enhancement of various human abilities. In the field of artificial intelligence, benefiting from high-quality annotated data (Deng et al., 2009; Wang et al., 2019), neural network models have been well evaluated and optimized specifically, leading to significant advancements across various tasks.

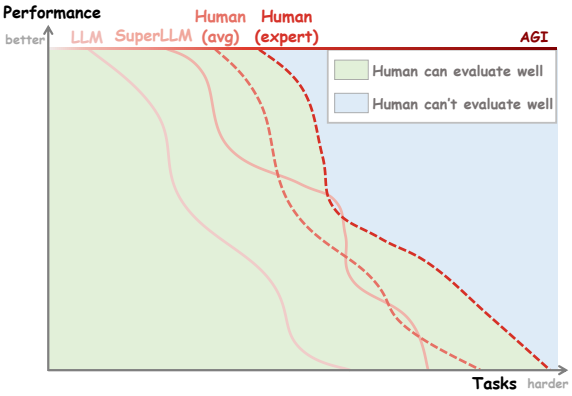


Figure 1: Schematic diagram of performance of human and models with varying task difficulty. Humans can only accurately assess problems within the scope of their capabilities.

Nowadays, advanced large language models (LLMs) (OpenAI, 2023), called SuperLLMs (Burns et al., 2023), have reached the level of human experts in certain tasks (Tedeschi et al., 2023; Lei et al., 2023; Pu et al., 2023; Webb et al., 2023). To further guide SuperLLMs towards AGI, evaluating them on a wider range of challenging tasks is of crucial importance.

However, building benchmarks for challenging tasks is especially intelligence-intensive and sometimes even difficult to ensure the accuracy of labels. As shown in Figure 1, annotating hard tasks requires scarce human experts and some tasks are even beyond the capability boundary of experts, such as "Is the Riemann hypothesis true?". So here comes the question: *How to evaluate LLMs accurately on benchmarks with poor supervision (with inaccurate or even no labels)?*

Some works have explored unsupervised (label-free) model evaluation, including using LLM-based

*Corresponding author.
[1]Our code and data have been released on https://github.com/ypw0102/PEEM.

Findings of the Association for Computational Linguistics ACL 2024, pages 11614–11627
August 11-16, 2024

pseudo labels (Bai et al., 2023; Li et al., 2023), evaluating with logits (Yu et al., 2022; Peng et al., 2024), and detecting conflicts with designed inputs (Fluri et al., 2023). Unfortunately, they are not proposed for evaluating SuperLLMs, presenting issues such as limited applicability across tasks, pseudo labels may still be inaccurate, logits of some LLMs are unavailable (e.g., GPT4), and dependence on prior knowledge. Overall, SuperLLMs are in eager need of an accurate, universal, and efficient poor-supervised evaluation paradigm.

Consequently, we propose the following theorem and prove it in Appendix §A.

Theorem 1. *For benchmark X lacking true labels Y, if reference model $\dot{\mathcal{M}}$ is independent of predictions $\{\hat{Y}^i\}_{i=1}^L$ of models $\{\mathcal{M}^i\}_{i=1}^L$ to be evaluated (Condition 1), and performs better than random guess (Condition 2), when the size of X is infinite (Condition 3), the following equation holds:*

$$\begin{aligned} &Cons(\dot{\mathcal{M}}(X), \hat{Y}^i) < Cons(\dot{\mathcal{M}}(X), \hat{Y}^j) \\ \Leftrightarrow\ &Cons(Y, \hat{Y}^i) < Cons(Y, \hat{Y}^j) \end{aligned} \tag{1}$$

where $Cons(\cdot)$ denotes mutual consistency.

Theorem 1 allows us to credibly evaluate and compare the capabilities of models (consistency between their predictions and true labels) without true labels Y.

LLMs themselves or humans can naturally serve as reference model $\dot{\mathcal{M}}$, but the conditions are not guaranteed to be sufficiently satisfied in real scenarios. Some LLMs may exhibit mutual affinity and tend to make the same predictions (e.g., Llama series), and the size of X is finite. Thus, we propose PEEM algorithm (**P**oor-supervised **E**valuation with **EM** algorithm) to alleviate these insufficiency. PEEM takes models predictions (with optional human annotations) as inputs, and calculates the weighted average consistency of certain model with reference models as the measurement of its capability. All the models to be evaluated are treated as reference models and assigned equal weights initially. Based on the EM algorithm, PEEM iteratively adjusts the weights of reference models during E-step, and filters reference models based on current weights during M-step to optimize the proxy optimization objective inferred from preliminary experiments. PEEM mitigates the limitation of sample size (Condition 3) by comprehensively utilizing the predictions of all models under evaluation. And it alleviates the dependency among model predictions (Condition 1) by optimizing the weights of reference models and filtering out the ones with high affinity tendency.

Our poor-supervised evaluation paradigm based on PEEM algorithm holds several advantages. **Efficient**: No additional model inference overhead is required except for model predictions. **Universal**: It can be applied across various tasks by proper definition of $Cons(\cdot)$, as we demonstrate in §5. **Accurate**: Experimental results show that PEEM aligns perfectly with true label evaluation results, achieving an average of 0.977 Pearson and 0.972 Spearman correlation coefficient.

Furthermore, we validate that PEEM with human annotations and model predictions, as opposed to the traditional consistency check between models and humans, allows for a more accurate evaluation results. More generally, PEEM has advanced the evaluation paradigm evolution from human-centric to human&model-centric, addressing the limitations of human evaluation capabilities.

Our contributions are summarized as follows:

1. We analyze and define the task of evaluating SuperLLMs as poor-supervised evaluation.
2. We theoretically provide a method and its conditions for poor-supervised evaluation.
3. We propose the PEEM algorithm to alleviate the insufficiency of the conditions for the introduced theorem under real-world scenarios by alternating between weight correction and reference model filtering.
4. We experimentally validate the efficiency, universality, and effectiveness of PEEM across regression, classification, and reasoning tasks.

2 Related Work

Surrounding our study, we discuss the researches on poor-supervised evaluation and model consistency, and briefly introduce the EM algorithm.

Poor-supervised Evaluation We refer to the process of evaluating models on benchmarks that contain inaccurately labeled or unlabeled data as poor-supervised evaluation, an area where many related efforts have been invested. Various directions have been explored to assessing model's capabilities on datasets without labels: examining distribution discrepancy (Yu et al., 2022; Deng et al., 2021) , relying on model confidence (Garg et al., 2021; Wang

et al., 2023a; Lu et al., 2023; Peng et al., 2024), calculating models' disagreements (Baek et al., 2022; Jiang et al., 2022; Chen et al., 2021; Jiang et al., 2021) and bucketing based on decision boundaries (Miao et al., 2023; Tu et al., 2023; Xie et al., 2023). However, these studies mainly focus on classification tasks and often require model logits, thus lacking universality and not being applicable to LLMs whose logits are unavailable. In addition, Fluri et al. (2023) and Jain et al. (2023) consider detecting conflicts with designed inputs to assess model's capabilities. However, the design of input format highly depends on prior knowledge and cannot be generalized across tasks. Conducting self-evaluation (Zheng et al., 2023) or peer evaluation (Li et al., 2023; Bai et al., 2023) with LLMs is a promising direction, but they have not addressed the fundamental issue of how to ensure the effectiveness of evaluations on benchmarks that exceed their own capabilities. Additionally, such methods require extra inference overhead. Overall, an accurate, universal, and efficient poor-supervised evaluation paradigm for SuperLLMs remains to be researched, for which we propose PEEM.

Model Consistency Consistency has long been studied for training models (Miyato et al., 2019; Chen et al., 2020), enhancing performance during inference (Wang et al., 2023b; Yao et al., 2023), and examining specific attributes such as reliability (Jang and Lukasiewicz, 2023) and hallucination (Ji et al., 2023). We investigate using mutual consistency between inaccurate predictions to achieve poor-supervised evaluation for SuperLLMs.

EM Algorithm The EM algorithm (Dempster et al., 1977) alternates between the Expectation step (E-step) and the Maximization step (M-step) to estimate parameters of models with latent variables. The E-step estimates the distribution of latent variables. During the M-step, parameters are updated to maximize the expected likelihood function. We use the EM algorithm to optimize the parameters of mapping from inter-prediction consistency to prediction-label consistency.

3 Preliminary

3.1 Task Definition

Given LLMs $\{\mathcal{M}^i\}_{i=1}^L$, datasets $\mathbb{D} : (X = \{x_i\}_{i=1}^N, Y = \{y_i\}_{i=1}^N)$ and predictions of $\mathcal{M}^i$ on X as $\hat{Y}^i = \{\hat{y}_j^i\}_{j=1}^N$, the consistency between $\{\hat{Y}^i\}_{i=1}^L$ and true labels Y, denoted as $\{Cons(\hat{Y}^i, Y)\}_{i=1}^L$, shortened to $B \in \mathcal{R}^L$, is the accurate measurement of the capabilities of these LLMs. Our research aims to assess LLMs when Y is unavailable. Therefore, we consider designing a mapping algorithm $\mathcal{F}$ that can map LLM predictions $\{\hat{Y}^i\}_{i=1}^L$ (with inaccurate human annotations if available) to capability vector $\hat{B} \in \mathcal{R}^L$, where $\hat{B}_i$ can substitute for B_i in evaluating the capabilities of model $\mathcal{M}^i$. We calculate the correlation coefficient between $\hat{B}$ and B to validate the effectiveness of $\mathcal{F}$. It should be noted that the calculation method of $Cons(\cdot)$ depends on the label domain, which we will introduce in §5.

3.2 Intuition of Theorem 1

We propose Theorem 1 to credibly implement poor-supervised evaluation, which is grounded in an intuition that is straightforward to understand: For the samples where the reference model predicts correctly, strong model tends to have higher consistency with the reference model compared to weak model, $Cons(\mathcal{M}^{strong}, \dot{\mathcal{M}}) > Cons(\mathcal{M}^{weak}, \dot{\mathcal{M}})$. While for the samples where the reference model predicts incorrectly, if the reference model does not tend to make the same predictions as either strong or weak models, the consistency of both strong and weak model with the reference model should be the same, $Cons(\mathcal{M}^{strong}, \dot{\mathcal{M}}) = Cons(\mathcal{M}^{weak}, \dot{\mathcal{M}})$. Therefore, across all the samples, $Cons(\mathcal{M}^{strong}, \dot{\mathcal{M}}) > Cons(\mathcal{M}^{weak}, \dot{\mathcal{M}})$.

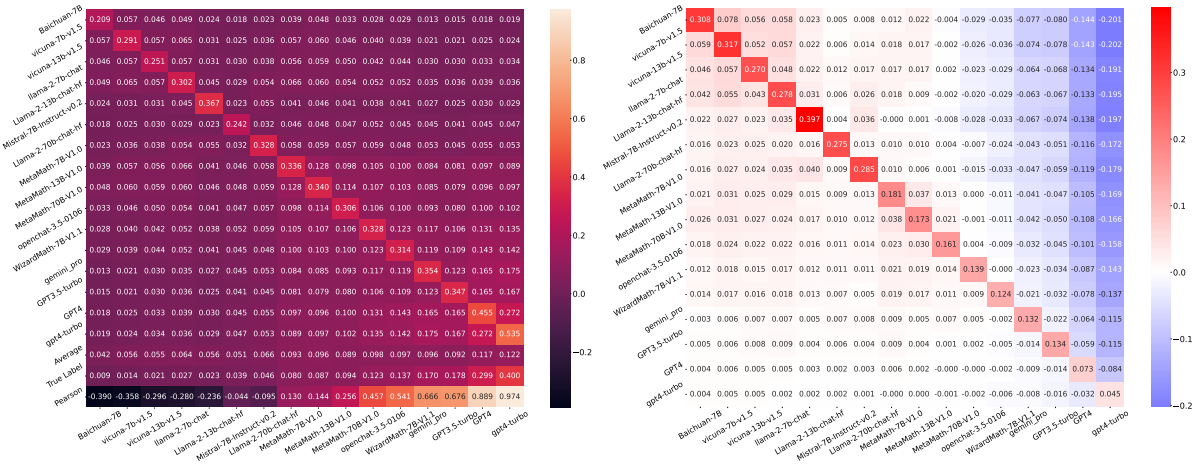
3.3 Visualization of Consistency and Affinity

Although Condition 2 can be assumed to be satisfied naturally, Condition 1 and 3 cannot be completely fulfilled in real scenarios as discussed before. To explore the impact of the insufficiency of Condition 1 and 3 in real scenarios, we visualize the consistency matrix C and the affinity matrix A of models to be evaluated. $C_{i,j} \in [0,1]$ denotes $Cons(\hat{Y}^i, \hat{Y}^j)$, the average consistency between the predictions of $\mathcal{M}^i$ and $\mathcal{M}^j$ on X. And $A_{i,j} \in [-1,1]$ reflects the affinity of $\mathcal{M}^i$ to $\mathcal{M}^j$, which can be calculated as follows:

$$A_{i,j} = \frac{C_{i,j}}{Sum(C_i)} - \frac{B_j}{Sum(B)} \quad (2)$$

A positive $A_{i,j}$ implies using $\mathcal{M}^i$ as the reference model to estimate $\hat{B}_j$ will overestimate the performance of $\mathcal{M}^j$ and vice versa.

We conduct this preliminary experiment on subset Precalculus of MATH (Hendrycks et al., 2021)



(a) Consistency Matrix (b) Affinity Matrix

Figure 2: Inter-model consistency matrix and affinity matrix on MATH-Precalculus dataset among 16 LLMs.

benchmark with 16 LLMs (See detailed setup in §5.1.1). As shown in Figure 2(a), the top 16 rows demonstrate the C, the third-to-last row calculates $\{Avg(C_i)\}_{i=1}^{L}$, the penultimate row shows B, while the i^{th} column of the last row represents the Pearson coefficient between C_i and B (the closer to 1, the stronger the linear correlation). From Figure 2(a), we can observe the following tendencies and corresponding insights. (1) The third-to-last and penultimate row show a positive correlation $\rightarrow$ ***Insight 1***: *stronger models tend to have higher consistency with other models*; (2) The last and penultimate row also show a positive correlation $\rightarrow$ ***Insight 2***: *stronger models' consistency with other models can better reflect the true performance of these models.* Considering that model predictions tend to converge towards the true labels and predictions of stronger models are closer to the true labels, the above two insights are natural and reasonable.

From Figure 2(b), we confirm that some LLMs belonging to the same series (Llama-2 series, MetaMath series) indeed exhibit mutual affinity, which makes their consistency biased to reflect their true performance. Furthermore, we observe ***Insight 3***: *weaker models tend to exhibit higher affinity.* We attribute this to the consistency between weak models and strong models underestimating the performance of the strong models, indirectly leading to a higher affinity among weak models. More generally, the proportion of samples correctly predicted by a weak model is too low, using it as a reference model may not suffice to differentiate between strong and weak models.

4 Methods

In this section, we propose multiple mapping algorithms $\mathcal{F}$ to obtain $\hat{B}$ based on the insights above, as shown in Figure 3. For each algorithm, we first calculate the consistency matrix C according to $\hat{Y}$ and task types, thus $\mathcal{F}: C \rightarrow \hat{B}$.

4.1 Naive Ensemble

Due to the lack of prior knowledge about the capabilities of models, a straightforward approach is to consider all the models as reference models and calculate the ensemble consistencies of certain model with other models to measure its capability:

$$\mathcal{F}_{ensemble}(C) = \{Avg(C_i)\}_{i=1}^{L} \tag{3}$$

Compared with randomly choosing certain model as reference model, $\mathcal{F}_{ensemble}$ can enlarge sample size by L times to alleviate the insufficiency of Condition 3 through ensemble, thus offering a more stable measurement result.

4.2 Weight Calibration

Building upon $\mathcal{F}_{ensemble}$, considering that C_i of stronger model $\mathcal{M}^i$ aligns better with B (***Insight 2***), we contemplate calibrating the ensemble weights based on the capabilities of the models through iteration.

Let α_j^k be the weight assigned to $C_{i,j}$ after iteration k, which is initialized as $1/L$. The capability estimation of $\mathcal{M}^i$ for iteration k, $\hat{B}_i^k$, is the calibrated average of its raw consistencies with all the

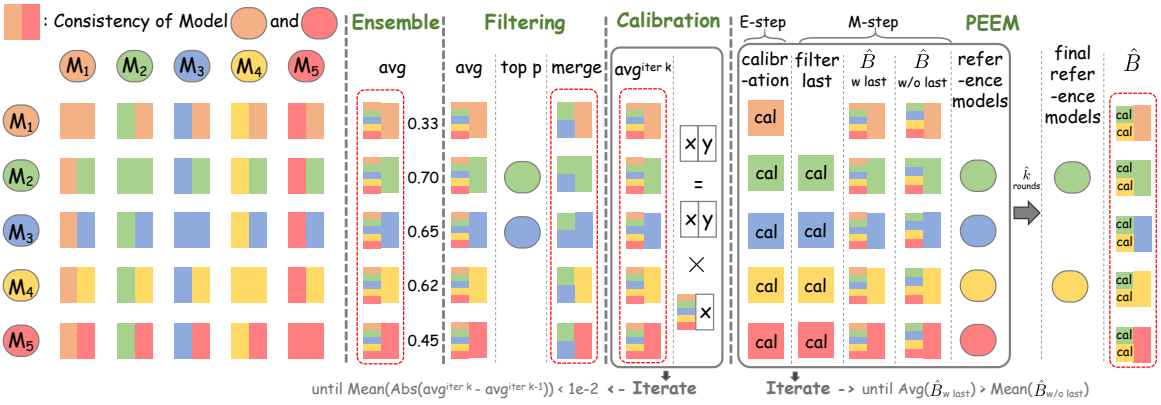


Figure 3: Overall illustration of our proposed methods.

models:

$$\hat{B}_i^k = \sum_{j=1}^{L} C_{i,j} \times \alpha_j^{k-1} \tag{4}$$

For each iteration k, we calibrate α_j^k according to $\hat{B}^k$ following ***Insight 2***:

$$\alpha_j^k = \frac{\hat{B}_j^k}{\sum_{i=1}^{L} \hat{B}_i^k} \tag{5}$$

Given the set of equations above, we look for the converging point after $\hat{k}$ iterations where the following equation holds:

$$\sum_{j=1}^{L} |\alpha_j^{\hat{k}} - \alpha_j^{\hat{k}-1}| < 1e-2 \tag{6}$$

Through the above process, we have managed to calibrate the ensemble weights using the capability estimation of the models, thereby obtaining a better algorithm $\mathcal{F}_{calibrate}$:

$$\mathcal{F}_{calibrate}(C) = \{\sum_{j=1}^{L} C_{i,j} \times \alpha_j^{\hat{k}}\}_{i=1}^{L} \tag{7}$$

4.3 Reference Model Filtering

To alleviate the impact of non-independent distributions between model predictions (insufficiency of Condition 1) shown in Figure 2(b), we consider filtering out models with strong affinity tendencies. ***Insight 3*** indicates that weak models generally have high affinity tendencies, thus we contemplate filtering them from reference models based on the estimation of their capabilities according to $\mathcal{F}_{ensemble}$ as follows:

$$\hat{B}^{ens} = \mathcal{F}_{ensemble}(C) \tag{8}$$

$$\mathcal{S}_{ref} = \arg\{i|\hat{B}_i^{ens} > Max(\hat{B}^{ens}) \times p\} \tag{9}$$

$$\mathcal{F}_{filter}(C) = \{Avg(\{C_{i,j}\}_{j\in\mathcal{S}_{ref}})\}_{i=1}^{L} \tag{10}$$

The hyperparameter p serves as a threshold for filtering out weak models whose performance is not up to par with the strongest model to a certain extent according to $\hat{B}^{ens}$. Through the above process, the weak model with high affinity tendencies can be filtered for calculating a debiased $\hat{B}$.

4.4 PEEM

Based on the discussions above, we consider designing an algorithm that can simultaneously leverage all three insights to mitigate the insufficiency of $Condition$ 1 and 3. To this end, we propose an EM-based algorithm $\mathcal{F}_{PEEM}$ as follows:

$$\mathcal{F}_{PEEM}(C) = \{\sum_{j=1}^{L} C_{i,j} \times \alpha_j^{\hat{k}} \times \beta_j^{\hat{k}}\}_{i=1}^{L} \tag{11}$$

We set $\beta_j \in \{0, 1\}$ to control whether $\mathcal{M}^j$ serves as a reference model and $\alpha_j \in [0, 1]$ to control how much does $\mathcal{M}^j$ contributes to $\hat{B}$.

Initialization. We initialize α_i^0 as $\frac{1}{L}$ and β_i^0 as 1.

Optimizing Objective. From ***Insight 1 and 2***, we deduce the following chain-of-thought: greater $Avg(C_i)$ $\overset{\text{Insight 1}}{\longleftrightarrow}$ stronger M_i $\overset{\text{Insight 2}}{\longleftrightarrow}$ better alignment between C_i and B. On this basis, $Avg(\hat{B})$ can serve as a proxy optimizing objective for $\hat{B}$ to attain a better approximation of B.

Expectation Step. In this step, we conduct $\mathcal{F}_{calibrate}$ among the reference model set constructed by β^{k-1} to obtain the calibrated expectation of α^k as follows:

$$\mathcal{S}_{ref}^{k-1} = \arg\{i|\beta_i^{k-1} = 1\} \tag{12}$$

$$\{\alpha_i^k\}_{i\in\mathcal{S}} = \mathcal{F}_{calibrate}(C_{i\in\mathcal{S}_{ref}, j\in\mathcal{S}_{ref}}) \tag{13}$$

Maximization Step. Further, we optimize β^k to maximize our objective $Avg(\hat{B}^k)$ as follows:

$$Index^k_{min} = \underset{i}{\text{argmin}}\ \alpha^k_i \tag{14}$$

$$\beta^k_i = \begin{cases} \beta^{k-1},\ i \neq Index^k_{min} \\ 0,\ i = Index^k_{min} \end{cases} \tag{15}$$

$$Avg(\hat{B}^k) = \frac{1}{\sum_{i=1}^{L}\beta^k_i}\sum_{i=1}^{L}\sum_{j=1}^{L} C_{i,j} \times \alpha^k_j \times \beta^k_j \tag{16}$$

Termination condition. We iteratively conduct the E-step and M-step for $\hat{k}+1$ rounds until the following formula holds true for the first time:

$$Avg(\hat{B}^{\hat{k}}) > Avg(\hat{B}^{\hat{k}+1}) \tag{17}$$

In summary, we alternately calibrate the model weights and filter out weak models with high affinity tendency according to the estimation under current parameter, iterating this process until the optimal proxy objective $Avg(\hat{B})$ is achieved.

5 Experiments

Since it is impossible to obtain accurate true labels for samples that human capabilities can not handle well, theoretically, we cannot directly measure the effectiveness of the proposed algorithms on such benchmarks. Thus we have designed a comparable experimental setup: we select benchmarks (MATH (Hendrycks et al., 2021) §5.1, USR (Mehri and Eskénazi, 2020) §5.2) that are currently considered fully manageable by human experts, and treat human predictions as the true labels. Under this setup, we validate whether $\mathcal{F}_{PEEM}(Cons(\text{LLMs}))$ can align well with $Cons(\text{LLMs, true label})$. Furthermore, MultiRC (Khashabi et al., 2018) §5.3 uses the predictions of single human annotator as human annotations and approximates the ensemble of predictions from multiple human annotators as true labels. We adopt this setting to verify if $\mathcal{F}_{PEEM}(Cons(\text{LLMs and human}))$ aligns better with $Cons(\text{LLMs, true label})$ than traditional evaluation paradigm $Cons(\text{LLMs, human})$.

5.1 Reasoning Task

We first validate the effectiveness of our proposed algorithms on MATH (Hendrycks et al., 2021) benchmark, which contains 7 subsets (Table 7), to assess the reasoning capabilities of LLMs.

5.1.1 Experimental Setup

We have selected 16 mainstream LLMs with a broad range of size and capabilities for the experiments involved in this paper, as shown in Table 6. To enhance the stability of evaluation, for each model $\mathcal{M}^i$ and sample x_j, we sample $T(\hat{Y}^i)$ (= 5 across our experiments) times at temperature 0.5 to attain predictions $\{\hat{y}^i_{j,k}\}_{k=1}^{T(\hat{Y}^i)}$ (Results of greedy search are also provided in §C.1). As the domain of true label y is discrete in MATH, we calculate $Cons(Y, \hat{Y}^i)$ as follows:

$$Cons(Y, \hat{Y}^i) = \frac{1}{NT(\hat{Y}^i)T(Y)}\sum_{j=1}^{N}\sum_{u=1}^{T(Y)}\sum_{v=1}^{T(\hat{Y}^i)} \mathbf{1}_{y_{j,u}=\hat{y}^i_{j,v}} \tag{18}$$

The inter-model consistency $Cons(\hat{Y}^i, \hat{Y}^j)$ can be similarly calculated. $T(Y) = 1$ as the label is unique for each x_i. To enhance the significance of the results, we randomly sample models at a ratio of $q_{model} = 0.7$ from 16 candidates to conduct experiments and take the mean of 500 outcomes to report. As for $\mathcal{F}_{filter}$, we set filtering threshold p as 0.9 where it can achieve the best performance (See Appendix §C.2 for results with varying p). The Pearson coefficient r_p can measure the degree of linear correlation, while the Spearman coefficient r_s can assess the monotonic relationship between two variables. We use them to measure the degree of alignment between $\hat{B}$ and B, which are separately obtained from our algorithm and the true labels.

5.1.2 Experimental Results

As shown in Table 1, our algorithms generally achieve correlation coefficient greater than 0.8 (0.9 for $\mathcal{F}_{PEEM}$) across most subsets, indicating that they can assess model capabilities well through mutual consistency in the absence of true labels.

In a horizontal comparison, $\mathcal{F}_{ensemble}$ ranks last as it only naively treat all the models as reference models. $\mathcal{F}_{calibrate}$ significantly improves the ordering of model capability (higher r_s) by adjusting the weights of models during ensemble. However, altering the original weights also leads to a slight decrease in the linear relationship (lower r_p). By filtering out weak models with high affinity tendency on the basis of $\mathcal{F}_{ensemble}$, $\mathcal{F}_{filter}$ achieves significant improvements on both r_s and r_p. Finally, by integrating the ideas of both weight adjustment and reference model filtering into EM algorithm and directly optimizing the proxy target $Avg(\hat{B})$,

Subset	IA		PA		GE		CP		AG		NT		PC		Average	
	r_p	r_s	r_p	r_s	r_p	r_s	r_p	r_s	r_p	r_s	r_p	r_s	r_p	r_s	r_p	r_s
Ensemble	.739	.791	.964	.992	.915	.970	.873	.983	.952	.994	.859	.969	.793	.866	.871	.938
Calibration	.784	.915	.941	.992	.883	.984	.850	.992	.921	.993	.846	.985	.826	.969	.864	.976
Filtering	.834	.886	**.997**	**.998**	**.986**	.990	.980	.994	.995	.999	.969	.983	.889	.931	.950	.969
PEEM	**.912**	**.958**	.994	.997	.978	.992	**.985**	**.996**	**.996**	**1.00**	**.977**	**.991**	**.955**	**.987**	**.971**	**.989**

Table 1: Model-level Pearson (r_p) / Spearman (r_s) correlations of different algorithms on 7 subsets of MATH.

q_{sample}	0.1		0.2		0.4		0.6		0.8		1	
	r_p	r_s	r_p	r_s	r_p	r_s	r_p	r_s	r_p	r_s	r_p	r_s
Ensemble	.824	.788	.867	.836	.890	.851	.901	.854	.902	.859	.909	.868
Calibration	.819	.805	.869	.894	.871	.984	.903	.883	.908	.885	.913	.883
Filtering	.852	.830	.898	.885	.920	.908	.929	.912	.933	.917	.937	.920
PEEM	**.878**	**.852**	**.921**	**.895**	**.952**	**.928**	**.957**	**.937**	**.961**	**.942**	**.966**	**.942**

Table 2: Model-level Pearson (r_p) / Spearman (r_s) correlations of different algorithms with different sample sampling ratios q_{sample} on USR benchmark.

$\mathcal{F}_{PEEM}$ achieves evaluation results that align perfectly with the true labels (0.971 r_p and 0.989 r_s).

Meanwhile, we notice that the performance of the proposed algorithms vary across different subsets. To this end, we compare the average accuracy of all the models $Avg(B)$, the mean of the absolute values of affinity matrix $Avg(Abs(A))$, and r_s between $\hat{B}$ from $\mathcal{F}_{PEEM}$ and B of each subset. We plot the 20^{th} power of r_s and $0.5 - Avg(Abs(A))$ for better observation. As shown in Figure 4, they exhibit a clear correlation in the graph. We speculate the reason is as follows: on subsets where the model accuracy ($Avg(B)$) is higher, their predictions tend to be consistent with the true label and thus show a weak affinity tendency ($Avg(Abs(A))$). Therefore, $Condition$ 2 can be well satisfied for $\mathcal{F}_{PEEM}$ to attain good performance (r_s), and vice versa. Inspired by this, we suggest that, where feasible, samples of moderate difficulty rather than those obviously beyond the model's capability should be chosen to evaluate the model, thereby making the evaluation results more accurate. Meanwhile, $\mathcal{F}_{PEEM}$ can be used to estimate whether samples are too difficult for the model during benchmark construction.

5.2 Regression Task

We select the USR benchmark (Mehri and Eskénazi, 2020) to validate our proposed algorithms within the context of regression task.

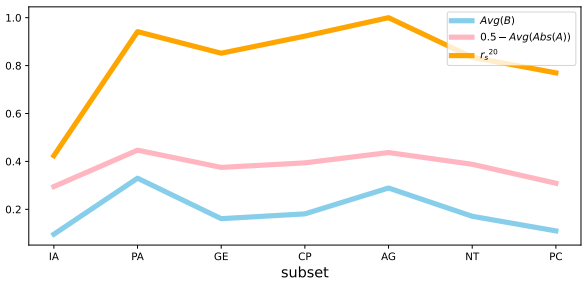


Figure 4: Comparisons between r_s of $\mathcal{F}_{PEEM}$, the mean of the absolute values of affinity matrix $Avg(Abs(A))$, and average accuracy of all the models $Avg(B)$. We plot the 20^{th} power of r_s and $0.5 - Avg(Abs(A))$ for easier observation.

5.2.1 Experimental Setup

USR is a dialog evaluation testbed requiring models to predict appropriate scores based on specific criteria for the provided dialogues. We follow Mehri and Eskénazi (2020); Liu et al. (2023) to calculate $Cons(Y, \hat{Y}^i)$ as shown below:

$$Cons(Y, \hat{Y}^i) = r_p(Y, \hat{Y}^i) \tag{19}$$

We choose the `Overall` criteria for experiments, which contains 360 samples. To validate the effectiveness of the proposed methods varying sample sizes, we conduct experiments with sample sampling ratios $q_{sample} \in [0.1, 0.2, 0.4, 0.6, 0.8, 1]$.

5.2.2 Experimental Results

As shown in Table 2, $\mathcal{F}_{PEEM}$ shows the best performance as expected, achieving 0.966 r_p and 0.942 r_s when $q_{sample} = 1$. As q_{sample}

q_{sample}	0.1		0.2		0.4		0.6		0.8		1	
	r_p	r_s	r_p	r_s	r_p	r_s	r_p	r_s	r_p	r_s	r_p	r_s
HUMAN	.965	.923	.981	.954	.990	.968	.992	.972	**.994**	.972	.994	.976
PEEM	**.978**	**.962**	**.986**	**.975**	**.991**	**.982**	**.993**	**.985**	.993	**.987**	**.994**	**.986**

Table 3: Model-level Pearson (r_p) / Spearman (r_s) correlations of different algorithms with different sample sampling ratios q_{sample} on MultiRC benchmark.

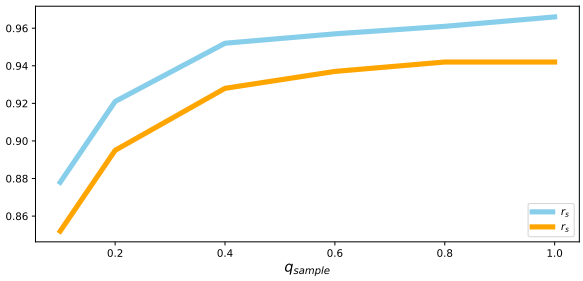


Figure 5: The trend of r_p and r_p of $\mathcal{F}_{PEEM}$ as q_{sample} changes.

gradually increases, the performance of all algorithms improves accordingly. This is natural since $Condition$ 1 is satisfied to a greater extent with the increase of sample size. We also notice that r_s and r_p exhibit logarithmic curves as q_{sample} increases (see Figure 5). This inspires us that the benefits of increasing sample size are greatest in the early stages. Therefore, when sample size is small (our experiments suggest 100 being a suitable threshold), we should try to collect samples as much as possible. However, when the number of samples is sufficiently large, we can afford to filter out some samples to save on inference resources, without significantly affecting the evaluation results.

5.3 Classification Task

To verify the effectiveness of $\mathcal{F}_{PEEM}$ within the context of classification task, we select MultiRC (Khashabi et al., 2018) benchmark for experiments.

5.3.1 Experimental Setup

MultiRC evaluates the reading comprehension ability of models by having them answer multiple-choice questions. Based on our previous observation that a large sample size can lead to performance gain saturation, we randomly select 200 samples to conduct experiments for saving inference costs with 16 LLMs. We use the macro-average $F1$ score to calculate $Cons(Y, \hat{Y}^i)$ following Khashabi et al. (2018), as shown below:

$$Cons(Y, \hat{Y}^i) = \frac{1}{NT(Y)T(\hat{Y}^i)} \sum_{j=1}^{N} \sum_{u=1}^{T(Y)} \sum_{v=1}^{T(\hat{Y}^i)} F1(y_{j,u}, \hat{y}^i_{j,v}) \tag{20}$$

Each sample in MultiRC is annotated by multiple individuals, and the ensemble result of these multiple annotations is considered the true label. The prediction made by a certain person is regarded as the human-level label $\hat{y}^{human}$. Traditionally, $Cons(\hat{y}^{human}, \hat{y}^i)$ is used to measure the capability of $\mathcal{M}^i$. Considering human as a model, we input the consistency matrix among $\{\hat{y}^i\}_{i \in \{human\} \cup \{1,...,L\}}$ into the proposed algorithm to verify whether our method can assist humans in better evaluating model capabilities compared to $Cons(\hat{y}^{human}, \hat{y}^i)$.

5.3.2 Experimental Results

As shown in Table 3, we find that $\mathcal{F}_{PEEM}$ surpasses traditional human-centric evaluation approach by an obvious margin, especially when q_{sample} is small. $\mathcal{F}_{PEEM}$ has advanced the evolution of evaluation paradigm from human-centric to human & model-centric paradigm, utilizing the capabilities of LLMs to compensate for the insufficiencies in human ability during evaluation.

6 Conclusions

In this paper, we propose PEEM, an algorithm for conducting poor-supervised evaluation without true labels for SuperLLMs. Comprehensive experiments validate the efficiency, effectiveness, and university of PEEM across 3 types of tasks with 16 mainstream LLMs. PEEM has advanced the evaluation paradigm evolution from human-centric to human&model-centric, alleviating the limitations of human capabilities for evaluating SuperLLMs. Our work is a preliminary exploration of evaluating SuperLLMs by analogizing existing LLMs and benchmarks, longing for further improvement in the future.

Limitations

From an objective perspective , we think there are two main limitations of this paper:

1. Since humans currently cannot construct benchmarks with accurate labels that exceed the boundaries of human capabilities, we can only conduct analogy experiments on datasets within the scope of human capabilities. Despite this, such an analogy is reasonable because changes in task difficulty do not affect the form of the experimental setup.

2. The experiments above have verified the effectiveness of $\mathcal{F}_{PEEM}$ in various tasks within closed label domain. Meanwhile, open-domain text generation is also an important aspect of LLM capability assessment. Unlike closed domain tasks holding a one-to-one relationship between input and output, open-domain tasks can have multiple appropriate answers for a given input. Therefore, evaluation based on consistency may not be suitable for open-domain evaluation and current mainstream methods depend on human-evaluator and LLM-evaluator (Liu et al., 2023). However, we still need a method to judge the capability of the LLM-evaluators when their capability surpass that of human-evaluator. The output of the LLM-evaluator is closed-domain, where we can use $\mathcal{F}_{PEEM}$ to carry out such evaluations, as demonstrated in §5.2. To summarize, $\mathcal{F}_{PEEM}$ can enhance the accuracy of model capability assessment in open-domain tasks by aiding in the selection of SuperEvaluators.

Ethics Statement

All of the datasets used in this study were publicly available, and no annotators were employed for our data collection. We confirm that the datasets we used did not contain any harmful content and was consistent with their intended use (research). We have cited the datasets and relevant works used in this study.

Acknowledgments

This work is supported by the Beijing Natural Science Foundation, China (Nos. 4222037, L181010).

References

Christina Baek, Yiding Jiang, Aditi Raghunathan, and J Zico Kolter. 2022. Agreement-on-the-line: Predicting the performance of neural networks under distribution shift. *Advances in Neural Information Processing Systems*, 35:19274–19289.

Yushi Bai, Jiahao Ying, Yixin Cao, Xin Lv, Yuze He, Xiaozhi Wang, Jifan Yu, Kaisheng Zeng, Yijia Xiao, Haozhe Lyu, et al. 2023. Benchmarking foundation models with language-model-as-an-examiner. *arXiv preprint arXiv:2306.04181*.

Collin Burns, Pavel Izmailov, Jan Hendrik Kirchner, Bowen Baker, Leo Gao, Leopold Aschenbrenner, Yining Chen, Adrien Ecoffet, Manas Joglekar, Jan Leike, et al. 2023. Weak-to-strong generalization: Eliciting strong capabilities with weak supervision. *arXiv preprint arXiv:2312.09390*.

Jiefeng Chen, Frederick Liu, Besim Avci, Xi Wu, Yingyu Liang, and Somesh Jha. 2021. Detecting errors and estimating accuracy on unlabeled data with self-training ensembles. *Advances in Neural Information Processing Systems*, 34:14980–14992.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. 2020. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR.

Arthur P Dempster, Nan M Laird, and Donald B Rubin. 1977. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society: series B (methodological)*, 39(1):1–22.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.

Weijian Deng, Stephen Gould, and Liang Zheng. 2021. What does rotation prediction tell us about classifier accuracy under varying testing environments? In *International Conference on Machine Learning*, pages 2579–2589. PMLR.

Lukas Fluri, Daniel Paleka, and Florian Tramèr. 2023. Evaluating superhuman models with consistency checks. *arXiv preprint arXiv:2306.09983*.

Saurabh Garg, Sivaraman Balakrishnan, Zachary Chase Lipton, Behnam Neyshabur, and Hanie Sedghi. 2021. Leveraging unlabeled data to predict out-of-distribution performance. In *International Conference on Learning Representations*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the MATH dataset. In *Proceedings*

of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual.

Neel Jain, Khalid Saifullah, Yuxin Wen, John Kirchenbauer, Manli Shu, Aniruddha Saha, Micah Goldblum, Jonas Geiping, and Tom Goldstein. 2023. Bring your own data! self-supervised evaluation for large language models. *CoRR*, abs/2306.13651.

Myeongjun Jang and Thomas Lukasiewicz. 2023. Consistency analysis of chatgpt. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 15970–15985. Association for Computational Linguistics.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Comput. Surv.*, 55(12):248:1–248:38.

Yiding Jiang, Vaishnavh Nagarajan, Christina Baek, and J Zico Kolter. 2021. Assessing generalization of sgd via disagreement. *arXiv preprint arXiv:2106.13799*.

Yiding Jiang, Vaishnavh Nagarajan, Christina Baek, and J. Zico Kolter. 2022. Assessing generalization of SGD via disagreement. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. 2018. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 252–262. Association for Computational Linguistics.

Bin Lei, Chunhua Liao, Caiwen Ding, et al. 2023. Boosting logical reasoning in large language models through a new framework: The graph of thought. *arXiv preprint arXiv:2308.08614*.

Ruosen Li, Teerth Patel, and Xinya Du. 2023. Prd: Peer rank and discussion improve large language model based evaluations. *arXiv preprint arXiv:2307.02762*.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: NLG evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 2511–2522. Association for Computational Linguistics.

Yuzhe Lu, Zhenlin Wang, Runtian Zhai, Soheil Kolouri, Joseph Campbell, and Katia Sycara. 2023. Predicting out-of-distribution error with confidence optimal transport. *arXiv preprint arXiv:2302.05018*.

Shikib Mehri and Maxine Eskénazi. 2020. USR: an unsupervised and reference free evaluation metric for dialog generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 681–707. Association for Computational Linguistics.

Shuyu Miao, Lin Zheng, Jingjing Liu, and Hong Jin. 2023. K-means clustering based feature consistency alignment for label-free model evaluation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023 - Workshops, Vancouver, BC, Canada, June 17-24, 2023*, pages 3299–3307. IEEE.

Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. 2019. Virtual adversarial training: A regularization method for supervised and semi-supervised learning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(8):1979–1993.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Ru Peng, Heming Zou, Haobo Wang, Yawen Zeng, Zenan Huang, and Junbo Zhao. 2024. Energy-based automated model evaluation. *arXiv preprint arXiv:2401.12689*.

Xiao Pu, Mingqi Gao, and Xiaojun Wan. 2023. Summarization is (almost) dead. *arXiv preprint arXiv:2309.09558*.

Simone Tedeschi, Johan Bos, Thierry Declerck, Jan Hajic, Daniel Hershcovich, Eduard H Hovy, Alexander Koller, Simon Krek, Steven Schockaert, Rico Sennrich, et al. 2023. What's the meaning of superhuman performance in today's nlu? *arXiv preprint arXiv:2305.08414*.

Weijie Tu, Weijian Deng, Tom Gedeon, and Liang Zheng. 2023. A bag-of-prototypes representation for dataset-level applications. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pages 2881–2892. IEEE.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Jiexin Wang, Jiahao Chen, and Bing Su. 2023a. Toward auto-evaluation with confidence-based category relation-aware regression. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023b. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference*

on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023. OpenReview.net.

Taylor Webb, Keith J Holyoak, and Hongjing Lu. 2023. Emergent analogical reasoning in large language models. *Nature Human Behaviour*, 7(9):1526–1541.

Renchunzi Xie, Hongxin Wei, Yuzhou Cao, Lei Feng, and Bo An. 2023. On the importance of feature separability in predicting out-of-distribution error. *CoRR*, abs/2303.15488.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *CoRR*, abs/2305.10601.

Yaodong Yu, Zitong Yang, Alexander Wei, Yi Ma, and Jacob Steinhardt. 2022. Predicting out-of-distribution error with the projection norm. In *International Conference on Machine Learning*, pages 25721–25746. PMLR.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *CoRR*, abs/2306.05685.

A Proof of Theorem 1

Definition 1. *Given models M^i, M^j under evaluation, reference model $\dot{M}$, benchmark $\mathbb{D} : (X = \{x_i\}_{i=1}^N, Y = \{y_i\}_{i=1}^N)$ and predictions of M^i on X as $\hat{Y}^i = \{\hat{y}_j^i\}_{j=1}^N$, the consistency between Model M^i and Ground Truth Y, denoted as $Cons(\hat{Y}^i, Y)$, is typically used to measure the performance of Model M^i. We prove that the following equation holds:*

$$\begin{aligned} &Cons(\dot{Y}, \hat{Y}^u) < Cons(\dot{Y}, \hat{Y}^v) \\ \Leftrightarrow\ &Cons(Y, \hat{Y}^u) < Cons(Y, \hat{Y}^v) \end{aligned} \tag{21}$$

when condition 1: Both $P(\dot{Y}|X), P(\hat{Y}^i|X)$ and $P(\dot{Y}|X), P(\hat{Y}^j|X)$ are independently distributed; condition 2: $\dot{M}$ performs better than random predictor; condition 3: $N \to \infty$.

Proof. We prove the theorem separately in the continuous and discrete label domains as follows:

Discrete Domain In discrete label domain, we define $Cons(A, B) = Avg(\mathbf{1}_{a_i=b_i} | a_i \in A, b_i \in B)$. Without loss of generality, we assume that the label domain has T possible values and the distribution of $P(\hat{Y}^i|X)$ are as follows:

$$P(\hat{y}_j^i = t|x_j) = \begin{cases} \sigma_j^i, & t = y_j \\ (1-\sigma_j^i) \times \lambda_j^{i,t}, & t \neq y_j \ (\sum_{t \neq y_j} \lambda_j^{i,t} = 1) \end{cases} \tag{22}$$

As $\dot{M}$ performs better than random predictor, $\mathbb{E}(\dot{\sigma}) > \frac{1}{T}$ holds. Based on the assumptions above, we can derive as follows:

$$\begin{aligned} &Cons(\hat{Y}^u, \dot{Y}) < Cons(\hat{Y}^v, \dot{Y}) \\ \Leftrightarrow\ &\sum_{i=1}^N Cons(\hat{y}_i^u, \dot{y}_i) < \sum_{i=1}^N Cons(\hat{y}_i^v, \dot{y}_i) \\ \Leftrightarrow\ &\sum_{i=1}^N \mathbf{1}_{\hat{y}_i^u = \dot{y}_i} < \sum_{i=1}^N \mathbf{1}_{\hat{y}_i^v = \dot{y}_i} \\ \Leftrightarrow\ &\sum_{i=1}^N \sigma_i^u < \sum_{i=1}^N \sigma_i^v \\ \Leftrightarrow\ &N\mathbb{E}(\sigma^u) < N\mathbb{E}(\sigma^v) \quad , according\ to\ condition\ 3 \\ \Leftrightarrow\ &\mathbb{E}(\sigma^u) < \mathbb{E}(\sigma^v) \end{aligned} \tag{23}$$

From another direction:

$$\begin{aligned} &Cons(\hat{Y}^u, \dot{Y}) < Cons(\hat{Y}^v, \dot{Y}) \\ \Leftrightarrow\ &\sum_{i=1}^N Cons(\hat{y}_i^u, \dot{y}_i) < \sum_{i=1}^N Cons(\hat{y}_i^v, \dot{y}_i) \\ \Leftrightarrow\ &\sum_{i=1}^N \mathbf{1}_{\hat{y}_i^u = \dot{y}_i} < \sum_{i=1}^N \mathbf{1}_{\hat{y}_i^v = \dot{y}_i} \\ \Leftrightarrow\ &\sum_{i=1}^N (\sigma_i^u \dot{\sigma}_i + \sum_{t \neq y_i} \lambda_i^{u,t} \dot{\lambda}_i^t) < \sum_{i=1}^N (\sigma_i^v \dot{\sigma}_i + \sum_{t \neq y_i} \lambda_i^{v,t} \dot{\lambda}_i^t) \\ \Leftrightarrow\ &N\mathbb{E}(\sigma^u \dot{\sigma} + \sum_{t \neq y_i} \lambda_i^{u,t} \dot{\lambda}_i^t) < N\mathbb{E}(\sigma^v \dot{\sigma} + \sum_{t \neq y_i} \lambda_i^{v,t} \dot{\lambda}_i^t) \\ &, according\ to\ condition\ 3 \\ \Leftrightarrow\ &\mathbb{E}(\sigma^u)\mathbb{E}(\dot{\sigma}) + (T-1)\mathbb{E}(\lambda^u)\mathbb{E}(\dot{\lambda}) \\ &< \mathbb{E}(\sigma^v)\mathbb{E}(\dot{\sigma}) + (T-1)\mathbb{E}(\lambda^v)\mathbb{E}(\dot{\lambda}) \\ &, according\ to\ condition\ 1 \\ \Leftrightarrow\ &\mathbb{E}(\sigma^u)\mathbb{E}(\dot{\sigma}) + (1-\mathbb{E}(\sigma^u))(1-\mathbb{E}(\dot{\sigma}))/(T-1) \\ &< \mathbb{E}(\sigma^v)\mathbb{E}(\dot{\sigma}) + (1-\mathbb{E}(\sigma^v))(1-\mathbb{E}(\dot{\sigma}))/(T-1) \\ \Leftrightarrow\ &(\mathbb{E}(\sigma^u) - \mathbb{E}(\sigma^v))(\mathbb{E}(\dot{\sigma}) - \frac{1-\mathbb{E}(\dot{\sigma})}{T-1}) < 0 \\ \Leftrightarrow\ &(\mathbb{E}(\sigma^u) - \mathbb{E}(\sigma^v))(T\mathbb{E}(\dot{\sigma}) - 1) < 0 \\ \Leftrightarrow\ &\mathbb{E}(\sigma^u) < \mathbb{E}(\sigma^v) \quad , according\ to\ condition\ 2 \end{aligned} \tag{24}$$

According to Eq 23 and Eq 24, Eq 21 holds in discrete label domain.

Continuous Domain In continuous label domain, we define $Cons(A, B) = -Avg(Abs(a_i - b_i) | a_i \in A, b_i \in B)$. Without loss of generality, we assume $\hat{y}_i^u = y_i + g_i^u$, where g_i^u is zero-mean Gaussian noise with variance as σ_i^u. Based on the assumptions above, we can derive as follows:

$$\begin{aligned} &Cons(\hat{Y}^u, \dot{Y}) < Cons(\hat{Y}^v, \dot{Y}) \\ \Leftrightarrow\ &\sum_{i=1}^N Cons(\hat{y}_i^u, \dot{y}_i) < \sum_{i=1}^N Cons(\hat{y}_i^v, \dot{y}_i) \\ \Leftrightarrow\ &\sum_{i=1}^N |\hat{y}_i^u - \dot{y}_i| > \sum_{i=1}^N |\hat{y}_i^v - \dot{y}_i| \\ \Leftrightarrow\ &\sum_{i=1}^N |g_i^u| > \sum_{i=1}^N |g_i^v| \\ \Leftrightarrow\ &N\mathbb{E}(|g^u|) > N\mathbb{E}(|g^v|) \quad , according\ to\ condition\ 3 \\ \Leftrightarrow\ &\mathbb{E}(\sigma^u) > \mathbb{E}(\sigma^v) \end{aligned} \tag{25}$$

Subset	IA		PA		GE		CP		AG		NT		PC		Average	
	r_p	r_s	r_p	r_s	r_p	r_s	r_p	r_s	r_p	r_s	r_p	r_s	r_p	r_s	r_p	r_s
Ensemble	.679	.818	.956	.993	.910	.960	.856	.966	.949	.994	.857	.969	.773	.867	.854	.938
Calibration	.754	.913	.938	.994	.893	.986	.835	.980	.922	.993	.835	.974	.827	.953	.858	.971
Filtering	.672	.881	.982	.996	.931	.988	.932	.985	.970	.998	.923	.974	.796	.930	.886	.965
PEEM	.779	.947	.976	.995	.935	.992	.931	.986	.971	.997	.924	.975	.751	.962	**.895**	**.979**

Table 4: Model-level Pearson (r_p) / Spearman (r_s) correlations of different algorithms on 7 subsets of MATH with greedy search.

Subset	IA		PA		GE		CP		AG		NT		PC		Average	
	r_p	r_s	r_p	r_s	r_p	r_s	r_p	r_s	r_p	r_s	r_p	r_s	r_p	r_s	r_p	r_s
Filtering $p = 0.8$	.831	.892	.987	.995	.966	.990	.951	.994	.983	.998	.943	.987	.886	.952	.935	.972
Filtering $p = 0.9$	.834	.906	.997	.998	.986	.990	.980	.994	.995	.999	.969	.983	.889	.939	**.950**	**.973**
Filtering $p = 1.0$	.786	.917	.995	.998	.972	.991	.964	.989	.995	.999	.945	.981	.815	.932	.925	.972

Table 5: Model-level Pearson (r_p) / Spearman (r_s) correlations of $\mathcal{F}_{filter}$ on 7 subsets of MATH with different values of p.

Model	Size (B)	Inference
Baichuan-7B	7	A100 40G
vicuna-7b-v1.5	7	A100 40G
llama-2-7b-chat	7	A100 40G
Mistral-7B-Instruct-v0.2	7	A100 40G
MetaMath-7B-V1.0	7	A100 40G
WizardMath-7B-V1.1	7	A100 40G
openchat-3.5-0106	7	A100 40G
vicuna-13b-v1.5	13	A100 40G
Llama-2-13b-chat-hf	13	A100 40G
MetaMath-13B-V1.0	13	A100 40G
Llama-2-70b-chat-hf	70	A100 40G
MetaMath-70B-V1.0	70	A100 40G
Gemini-pro	-	API
gpt-3.5-turbo (0613)	-	API
gpt-4 (0613)	-	API
gpt-4-turbo (1106)	-	API

Table 6: Statistics of LLMs involved in the experiments.

From another direction:

$$
\begin{aligned}
& Cons(\hat{Y}^u, \dot{Y}) < Cons(\hat{Y}^v, \dot{Y}) \\
\Leftrightarrow & \sum_{i=1}^{N} Cons(\hat{y}_i^u, \dot{y}_i) < \sum_{i=1}^{N} Cons(\hat{y}_i^v, \dot{y}_i) \\
\Leftrightarrow & \sum_{i=1}^{N} |\hat{y}_i^u - \dot{y}_i| > \sum_{i=1}^{N} |\hat{y}_i^v - \dot{y}_i| \\
\Leftrightarrow & N\mathbb{E}(|\hat{y}^u - \dot{y}|) > N\mathbb{E}(|\hat{y}^v - \dot{y}|) \\
& , according\ to\ condition\ 1 \\
\Leftrightarrow & N * (\mathbb{E}(\sigma^u) + \mathbb{E}(\dot{\sigma})) > N * (\mathbb{E}(\sigma^v) + \mathbb{E}(\dot{\sigma})) \\
& , according\ to\ condition\ 2 \\
\Leftrightarrow & \mathbb{E}(\sigma^u) > \mathbb{E}(\sigma^v)
\end{aligned}
\tag{26}
$$

Subset	Abbreviation	Num
Intermediate Algebra	IA	903
Prealgebra	PA	871
Geometry	GE	479
Counting & Probability	CP	474
Algebra	AG	1187
Number Theory	NT	540
Precalculus	PC	546

Table 7: Statistics of MATH's subsets.

According to Eq 25 and Eq 26, Eq 21 holds in discrete label domain. □

B Detailed Statistics

We conduct experiments with 16 mainstream LLMs as shown in Table 6. The detailed statistics of MATH benchmark are shown in Table 7.

C Further Experimental Results

C.1 Results with Greedy Search

Apart from the main results of the proposed algorithms with random sampling at temperature as 0.5 and sampling times as 5, we also show the results with greedy search as shown in Table 4. PEEM performs best as expected. Compared with random sampling at sampling times as 5, the correlations between $\hat{B}$ and B decrease a little bit across all the algorithms. We believe this is because multiple sampling can eliminate the noise impact brought by single sampling.

C.2 Hyperparameter Analysis

We have explored the effectiveness of $\mathcal{F}_{filter}$ under different values of p. As shown in Table 5, $\mathcal{F}_{filter}$ performs relatively stably across different p-values, reaching its best when $p = 0.9$.