IDGen: Item Discrimination Induced Prompt Generation for LLM Evaluation

Fan Lin[1,2]*, **Shuyi Xie**[2*], **Yong Dai**[2*], **Wenlin Yao**[2], **Tianjiao Lang**[2], **Yu Zhang**[1†]
[1]SouthEast University, Nanjing, China, [2]Tencent, Shenzhen, China

Abstract

As Large Language Models (LLMs) grow increasingly adept at managing complex tasks, the evaluation set must keep pace with these advancements to ensure it remains sufficiently discriminative. Item Discrimination (ID) theory, which is widely used in educational assessment, measures the ability of individual test items to differentiate between high and low performers. Inspired by this theory, we propose an ID-induced prompt synthesis framework for evaluating LLMs to ensure the evaluation set can continually update and refine according to model abilities. Our data synthesis framework prioritizes both breadth and specificity. It can generate prompts that comprehensively evaluate the capabilities of LLMs while revealing meaningful performance differences between models, allowing for effective discrimination of their relative strengths and weaknesses across various tasks and domains. To produce high-quality data, we incorporate a self-correct mechanism into our generalization framework, and develop two models to predict prompt discrimination and difficulty score to facilitate our data synthesis framework, contributing valuable tools to evaluation data synthesis research. We apply our generated data to evaluate five SOTA models. Our data achieves an average score of 51.92, accompanied by a variance of 10.06. By contrast, previous works (i.e., SELF-INSTRUCT and WizardLM) obtain an average score exceeding 67, with a variance below 3.2. The results demonstrate that the data generated by our framework is more challenging and discriminative compared to previous works. We will release a dataset of over 3,000 carefully crafted prompts to facilitate evaluation research of LLMs. [3]

1 Introduction

The rapid advancement of LLMs, such as OpenAI's ChatGPT, Anthropic's Claude [1], and Facebook's LLaMA series [2, 3], has revolutionized the field of Natural Language Processing (NLP) in recent years. Model evaluation plays a crucial role in the development of LLMs, as it guides the iterative improvements during training, enables the selection of the best model variations, and facilitates their deployment in real-world applications [4, 5]. Recognizing the importance of model evaluation, researchers have made great efforts to create comprehensive benchmarks. Many of these benchmarks consist of multiple-choice questions in English [6, 7], as the results are easily obtainable through string matching. Some researchers [8] have extended these datasets to non-English languages, adapting the content to new linguistic and cultural contexts through translation. These datasets often result from either extensive public data collection or through manual or model-assisted data synthesis processes.

*Equal contribution. Work done during the internship of Lin at Tencent.
†Corresponding author.
[3]Code and data are available at `https://github.com/DUTlf/IDGen.git`

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

 https://doi.org/10.52202/079017-2809

Despite these advances, existing evaluation frameworks exhibit crucial limitations, particularly in their ability to discriminate between LLMs of varying capabilities. The predominant use of multiple-choice questions restricts the evaluation to specific competencies, potentially overlooking the full generative potential of LLMs, including their instruction-following ability. Merely translating prompts from one language to another language may not adequately demonstrate a model's proficiency within a specific cultural context. Furthermore, current generation methods lack a comprehensive mechanism to ensure the correctness of the generated questions, which is especially important for producing mathematic questions.

More importantly, the evaluation set should evolve adaptively as LLMs' abilities improve to ensure it remain sufficiently discriminative. As LLMs become more capable of handling increasingly complex tasks, the evaluation set must keep pace with these advancements. Static evaluation sets may be ineffective in differentiating between the performance of various LLMs. To maintain the discriminative power of the evaluation set, it is essential to continually update and refine the questions and tasks according to model abilities. This involves incorporating new challenges that push the boundaries of LLMs' abilities, such as more difficult reasoning, deeper understanding of context, and generating coherent responses to complex instructions. By adaptively updating the evaluation set in the development of LLMs, we can ensure that the benchmarks keep providing valuable insights into the strengths and weaknesses of different models.

To address these challenges, we propose a robust framework to produce high-quality, discriminative test data that evolves in alignment with advancements in LLM capabilities. Our framework is inspired by Item Discrimination (ID) Theory [9] that is introduced to assess how well individual questions (items) on a test distinguish between students who perform well on the overall test and those who do not. We adopt ID Theory to ensure each test question's effectiveness in differentiating between higher and lower-ability LLMs. Our framework can generate open-ended questions automatically in both English and Chinese, aimed at capturing a wide spectrum of tasks. Central to our approach is the application of discriminative techniques that enhance the test sets' ability to distinguish between different levels of language understanding, thereby allowing for a more precise evaluation of LLM performance. To achieve this goal, we also introduce two key metrics: question discriminative power and question difficulty, and train corresponding models to measure them. Additionally, we establish an iterative verification process to guarantee the logical soundness and precision of our questions. This multi-round iterative process can better enhance the usability of questions with logical coherence.

Our contributions can be summarized as follows:

- We propose a framework for data production and generalization that enables the rapid and high-quality creation of test datasets capable of effectively testing and differentiating LLMs.
- We innovatively adopt discrimination as the guiding principle for data production and generalization, employing rigorous data correction methods throughout the entire data production process to ensure the generated data has high usability and quality.
- We release a comprehensive set of over 3,000 questions, created and refined through our rigorous iterative verification process, to support and enrich the community's resources for LLM evaluation.
- We develop and train two models to measure question discriminative power and difficulty, which we have made available to the open-source community.

2 Method

In this section, we demontstrate our generalization framework in Figure 1. Assuming We have meticulously handcrafted a batch of high-quality seed data, the first thing is to exploit "instruction gradient"[4], i.e., specially designed rules from the instruction perspective, to generalize the questions (in Section 2.1). Subsequently, we employ "response gradient" to generalize questions, where the "gradient" refers to the rules for generalizing questions based on LLMs' responses (in Section 2.2). Next, we discuss a self-correct method to rectify the generalization questions, enhancing the usability

[4]Inspired by [10], the generalization of instruction data is similar to forward propagation in gradient descent while generalizing instructions by asking questions about the model's response is similar to backpropagation. Therefore, we name our methods as "instruction gradient" and "response gradient" respectively.

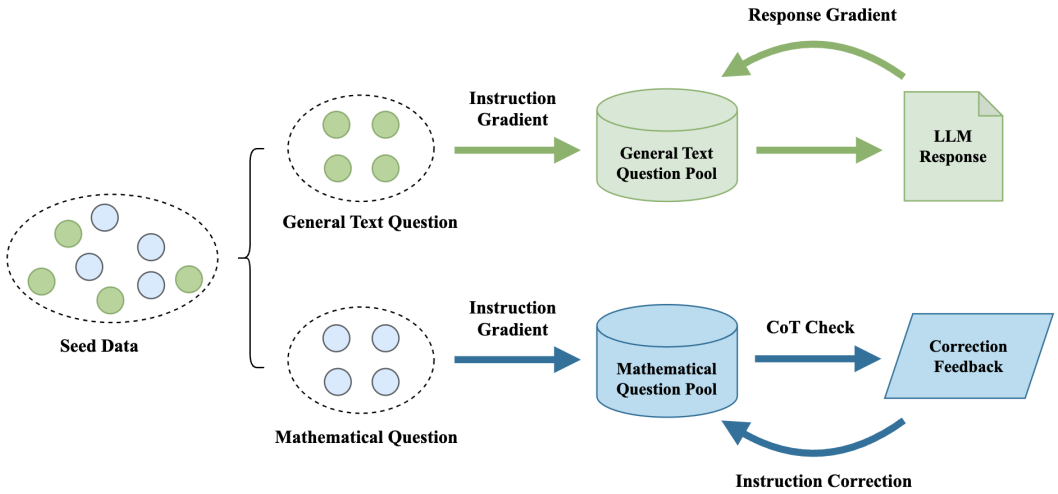

Figure 1: **Self-Correct Instruction Generalization Framework with "Instruction Gradient"**. Firstly, we handcraft a batch of seed data, dividing it into math category and general text category. Next, we generate a batch of dataset through "instruction gradient". For instructions in the general text category, we generate responses using a LLM, then generate new instructions through "response gradient", i.e., propose new questions based on the response. For problems in the math category, we check them through CoT check, and apply self-correct according to the CoT check's feedback.

of these data (in Section 2.3). Finally, we illustrate how we get high-quality answers from LLMs (in Section 2.4). To ensure the discrimitive power of the generation evaluation set, we propose to train an discrimination estimation model and an difficulty estimation model to formulate two metrics (in Section 2.5 and 2.6).

2.1 Data Generalization Based on "Instruction Gradient"

From the perspective of instruction, we aim to design constraints to guide the generated content, ensuring the generated questions adhere to specified content and also possess diversity and distinctiveness. Inspired by previous work, we refer to this feedback from the instruction perspective as "instruction gradient". We apply Hunyuan for data generalization[5] A.2. Since different types of data require distinct generalization techniques, we create various methods tailored to different categories of data [11]. We systematically develop several strategies that enhance both the difficulty and the discriminative power of the generated questions. In our study, we delineate 12 strategies tailored for addressing general text questions, such as "restricting the language used in responses", and formulate 8 distinct strategies for tackling mathematical questions, including "introduce additional variables". A comprehensive enumeration of these methodologies is presented in Appendix Table 5. In the data production process, for general text questions, we select 1-3 suitable generalization strategies. This approach aims to increase the complexity and differentiation of the generated questions, making them richer and more diverse. In contrast, for mathematical questions, we randomly select a single strategy. This choice helps to minimize the risk of generating unusable questions and ensures consistency in the problem generation process.

2.2 Instruction Generalization Reliant on "Response Gradient"

Generalizing questions from seed data based on the "instruction gradient" restricts the diversity and confines Especiallynt to specific topics. To enhance the diversity of general evaluation questions, we adopt a two-pronged approach. Firstly, we ensure overall diversity by expanding the variety of seed data. Secondly, we amplify question diversity by leveraging the "response gradient."

[5]Unless otherwise specified, all data in this document are generated by Hunyuan (Hunyuan-standard), which is a Large Language Model developed by Tencent. Additionally, data generated using other LLMs are also employed, and the relevant experiments and analyses are provided in the Appendix

https://doi.org/10.52202/079017-2809

For general text questions, we rephrase the question based on the response from the LLM. Specifically, we append a brief instruction to the question, which serves to guide the LLM in generating responses with more comprehensive information. After acquiring additional information, we generate new questions based on them. However, to ensure the difficulty and discrimination of the data, we embed a reference question in the prompt. We present the instruction that guides more information for the LLM and the prompt rephrasing questions based on response information in Appendix Table 6 and Table 7.

For example, for the question "How can NLP technology be used to detect and prevent the spread of fake news?", using the instruction gradient for generalization, we can obtain a new question "List three specific methods to detect and prevent the spread of fake news using NLP technology and explain their principles," which still revolves around the original question for expansion or transformation. To address this, we consider discarding the original question and using the LLM-generated response as information or knowledge. At this point, we only generate questions based on a piece of text, and the questions may become more interesting based on the content of the response. In the above example, we could generate a new question "What NLP tasks are typically addressed by fact-checking and source analysis techniques?"

2.3 Evaluating Question Usability

Assessing the Usability of General Text Questions Inspired by the methodologies outlined in [10], we craft a comprehensive set of evaluation criteria encompassing safety, neutrality, integrity and feasibility. These criteria are important in assessing the suitability of general text questions for our purposes. The detailed descriptions of these evaluative measures are presented in Appendix Table 8. We consider a question to be unusable if it fails to meet any of these criteria.

CoT Check for Mathematical Questions For mathematical questions, it is insufficient to estimate whether the generalization question is reasonable or not using a simple instruction for an LLM. As depicted in Figure2, consider the question "There are ten red, yellow, and blue balls in a box. You wish to draw a ball at random from the box. What are the chances of drawing a red ball?". The generated question includes two conditions, "totaling 30 balls" and "the probability of drawing a yellow ball is 1/4", which leads to a result of 7.5 yellow balls. This result contradicts common sense because there should not be a "half" yellow ball. Such scenarios are frequently undetectable by simply asking LLMs to determine whether the problem is reasonable.

Inspired by CoT (Chain of Thought) [12], we come up with a CoT-based approach to check whether generated questions are reasonable or not. Specially, we start with the concepts and move on to analyze each element of the problem, ensuring the rationality and precision of mathematical questions by assessing logical connections, solvability, and meticulously examining assumptions and calculation outcomes in the present context. The details are depicted in Appendix Table 10. Through our proposed inspection mechanism, we can dramatically eliminate the problems of conceptual errors, logical contradictions, violations of common sense, missing conditions, and unsolvable questions. In the example shown in Figure 2, we use Hunyuan to assess the reasonableness of the question, which successfully identifies the unreasonableness of the problem and corrects it based on the assessment process. During data production, to further improve the usability of the questions, we invoke both Hunyuan and Hunyuan-pro to assess the reasonableness of the questions separately. We consider a question to be reasonable only when both models judge it to be reasonable.

2.4 Acquiring reference answers

To ensure the highest quality of responses for general inquiries, we adopt a sophisticated multi-model strategy. We use five SOTA LLM models: Hunyuan, GPT-4, GPT4-Turbo, Wenxin 4, and Qwen[6] to generate preliminary answers independently. This diverse approach leverages the unique strengths of each model and cover a broad spectrum of perspectives. For general text questions, inspired by [13], we design each response from the following perspectives: Safety (0-30 points), Correctness (0-10 points), Relevance (0-10 points), Comprehensiveness (0-10 points), Readability (0-20 points), Richness (0-10 points), and Humanization (0-10 points). Hunyuan scores each response according to

[6]In this document, GPT-4 refers to gpt-4-32k-0613, GPT-4 Turbo refers to gpt-4-turbo-2024-04-09, Wenxin 4 refers to ERNIE-Bot 4.0, Qwen refers to Qwen-Max, Claude 3 refers to claude-3-opus-20240229.

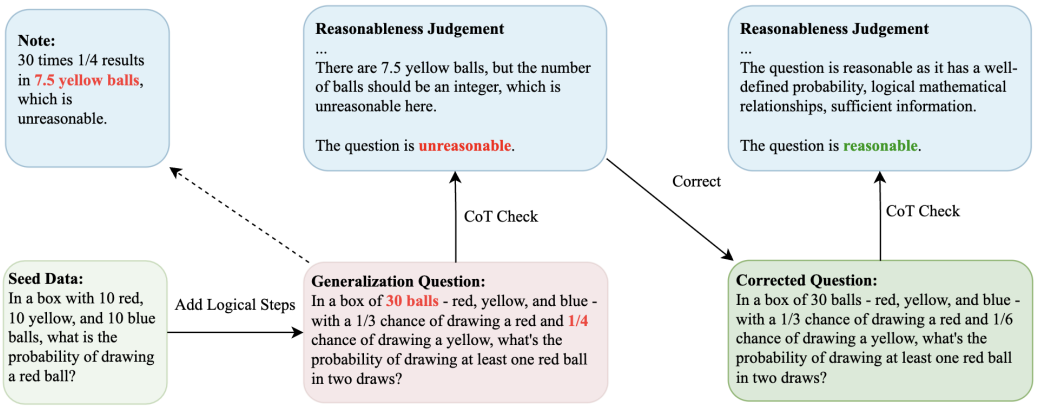

Figure 2: Chain of Thought Check Illustrated with a Mathematical Question Example

these criteria to maintain a high standard of consistency and fairness. The response with the highest score from Hunyuan is then selected as the reference answer and is used to establish a benchmark.

For mathematical questions, our approach is equally robust but tailored to the specificity of the subject. The most accurate response is determined through a collective voting mechanism[7] involving three models: Hunyuan, GPT-4 Turbo, and Qwen. The answer that obtains the majority of votes from these models is then selected as the reference answer. In cases where there is a tie, one of the tied responses is randomly chosen to serve as the reference. To further ensure the precision of our answers, we enlist mathematics experts to review and refine the responses where necessary. This step is crucial to validate the accuracy and dependability of the answers we provide.

2.5 Discrimination Estimation Model

To facilitate data synthesis and ensure new data are discriminative enough, we train a model to measure discrimination of each data instance. Each training instance includes prompts and its label discrimination indexes. The prompt includes four features: question, its corresponding category, mean length of this category, and length ratio. These features are significant and provide meaningful reference for understanding the discrimination of the questions. We apply a five-point rating system to score each response from different models and obtain the discrimination indexes. The specific scoring criteria can be seen in Table 1.

Table 1: Score Evaluation Criteria.

Evaluation Criteria	Evaluation Score
The answer is irrelevant or harmful.	0
The answer is wrong or contains factual errors.	1
The answer is correct but the process has flaws.	2
The answer is right.	3
The answer exceeds expectations.	4

Refer to the discrimination indexes proposed by T.L.Kelley [15] in education studies, we design a calculation formula for discrimination indexes by utilizing the evaluation data derived from several models including GPT-4, ChatGPT, Wenxin 4, and Qwen. Regarding the same question, arrange each model's average score in a descending order. The average score for the top 50% is denoted as PH, while the average score for the bottom 50% is indicated as PL. The computation of the discrimination indexes is articulated by the following formula:

[7]We hope to select high-quality responses as reference answers as much as possible. Related work[14] studies the theoretical basis of the collective voting mechanism and discusses the impact of different voting methods on social welfare. Inspired by this, we introduce a "collective voting mechanism" to select reference answers by comparing and voting among multiple responses.

https://doi.org/10.52202/079017-2809

$$PH = \frac{\sum_{i=1}^{N/2} \sum_{k=1}^{M} \text{score}_{ik}}{\frac{N}{2} * M} \tag{1}$$

$$PL = \frac{\sum_{i=\frac{N}{2}+1}^{N} \sum_{k=1}^{M} \text{score}_{ik}}{\frac{N}{2} * M} \tag{2}$$

$$\text{discrimination\_indexes} = \frac{PH - PL}{\text{max\_score}} \tag{3}$$

where N is the number of models, M is the total number of evaluators, $score_{ik}$ is the k-th evaluator's score for the i-th evaluation model's answer, and max_score is the highest score of the evaluation (in our scoring system, the max_score is 4). We map the discrimination indexes to four levels: "Low" for values less than or equal to 0.1, "Relatively Low" for values greater than 0.1 but less than or equal to 0.15, "Relatively High" for values greater than 0.15 but less than or equal to 0.25, and "High" for values greater than 0.25. The threshold here is estimated based on the distribution of 100,000-level evaluation data.

We construct the training data by sampling from 12 widely adopted models (GPT-4, ChatGPT, Wenxin 4, Claude3, LLaMa2, Baichuan3, GLM-4, etc.). A training sample includes information such as the question, category, reference answer, and the ratio of the question length to the average length of its category, etc. The expected label is a discrimination level label ranging from 0-3, which implies superior discrimination when the number is high. Then, Baichuan2-13B is used as the backbone to be supervised and finetuned as a discrimination model.

To more accurately obtain the discriminative power of the dataset, we calculate the discrimination indexes through manual annotation. Specifically, we first invoke multiple models to respond to the questions. Then we engage relevant experts to score the responses of various models according to Table 1. Subsequently, we calculate the discrimination indexes for each sample using Formula 3 and then determine the average value across all samples to obtain the discrimination indexes for this batch of data.

2.6 Difficulty Estimation Model

In our research, we utilize the "difficulty level" metric to assess a dataset's ability to differentiate various model by categorizing data into varying levels of difficulty. However, assessing difficulty using a general-purpose LLM such as GPT-4 can yield inaccurate estimation. Moreover, manually annotating the difficulty level of each instance is time-consuming and labor-intensive, and there's often a discrepancy between the difficulty perceived by humans and the difficulty perceived by models. To address these challenges, we have developed a specialized model designed specifically to evaluate the difficulty of each question. We train this model using a dataset compiled from the evaluation results of various LLMs, similar to those used in training our discrimination estimation model. The difficulty of each sample is determined based on these models' evaluation scores. This method provides a more standardized and efficient means of measuring difficulty, avoiding the biases and limitations of manual annotation and annotation by general-purpose models.

$$\text{difficulty\_score} = \text{max\_score} - \frac{\sum_{l=1}^{N} \sum_{j=1}^{M} \text{score}_{lj}}{M * N} \tag{4}$$

Where N is the number of evaluation models, M is the total number of evaluators, and $score_{ij}$ is the j-th evaluator's score for the i-th evaluation model's answer. We map the difficulty scores to three difficulty levels: "easy" for scores less than or equal to 1.5, "medium" for scores greater than 1.5 but less than or equal to 2.5, and "hard" for scores greater than 2.5. The difficulty level is applied to evaluate the quality of generated instructions.

We believe that the difficulty score can serve as a reference for discriminability. In addition, a high difficulty score for a question does not necessarily mean that it is more discriminative. For example, for a question with a max score of 3, if the evaluation scores are both 0 and 0, according to the formula, its difficulty score is 3, and the discrimination score is 0, meaning that the question is very

difficult, and the LLMs cannot answer it correctly, so the question is not discriminative. However, if the evaluation scores are 0 and 3, we can calculate that its difficulty score is 1.5, and the discrimination score is 1, indicating that the question can effectively distinguish the level of LLMs.

We propose a difficulty estimation model by fine-tuning BaiChuan2-13B pretrained model. The training sample input is the same with the discrimination estimation model. The output is 1-3, representing the difficulty level, and the training instruction is changed to estimate the difficulty of the problem. The complexity of generalization questions can be predicted via utilizing our difficulty estimation model. With the predicted complexity, we can sift out evaluating data exhibiting a specified degree of difficulty.

In order to obtain a more accurate measure of the difficulty of the dataset, we calculate the difficulty scores through manual annotation. After obtaining the annotators' scores for the responses of various models to the questions, we can calculate the difficulty score for each sample using Formula 4. By calculating the average value of the difficulty scores for all samples in the dataset, we obtain the difficulty score for these samples.

3 Experiment

In this section, we first introduce the experimental setup, including the baselines and the seed data. Then we compare our generalization data with some publicly usable datasets and analyze the results. Subsequently, we assess the usability of our data, as well as the discrimination indexes and difficulty score, and provide relevant analysis. Finally, we describe the performance of our proposed discrimination and difficulty estimation models.

3.1 Experiment setting

Baselines (1) SELF-INSTRUCT [16]: it generates approximately 82k instances from 175 human-created handwritten instructions.

(2) Instruction Tuning with GPT-4 Dataset [8]: in this task, GPT-4 is used to generate responses to the 52k English data from Alpaca dataset. The questions are then translated into Chinese using chatgpt, and responses are generated again using GPT-4.

(3) WizardLM [17]: it leverages the ChatGPT API to generate 250k instructions based on the training data from Alpaca Dataset.

Seed Data We establish a dataset comprising 6,000 instances by employing human annotators, which consists of Chinese and English subsets. The Chinese subset[11] is composed of approximately 5,000 instances, while the English subset contains 1,000 instances. The English instances include 175 sourced from the SELF-INSTRUCT dataset [16] and the remainder from the Alpaca dataset [18]. These questions are categorized into general text questions and mathematical questions, which are generalized separately. Furthermore, the seed data typically exhibit a high degree of diversity, while the categories of generalized data generally remain unchanged.

3.2 Comparison to Public Datasets

Discrimination indexes and difficulty score analysis Including the first three baselines that have already been introduced, we have also incorporated other datasets:

(1) SELF-INSTRUCT_seed_data: 175 seed data used to generate the SELF-INSTRUCT dataset.

(2)SELF-INSTRUCT-Ours: the dataset created by generalizing the 175 seed data points from the SELF-INSTRUCT dataset using our proposed method.

(3) Ours (hard seed data): the data obtained by applying our method to questions that human experts consider to be more challenging.

We sample responses from GLM-4, GPT-4 Turbo, GPT-4, Claude 3, and Qwen. We ask 104 domain experts to score the responses from each model according to the criteria outlined in Table 1 and calculate the discrimination and difficulty. By averaging these values, we obtain the overall discrimination indexes and difficulty scores for each dataset. The results are presented in Table 2.

 https://doi.org/10.52202/079017-2809

Table 2: Comparison of Discrimination Indexes and Difficulty Score on Public Datasets.

Dataset	Discrimination Indexes	Difficulty Score
WizardLM	0.140	1.235
Instruction Tuning with GPT-4	0.098	1.215
SELF-INSTRUCT_seed_data	0.061	1.146
SELF-INSTRUCT	0.109	1.319
SELF-INSTRUCT-Ours	0.137	1.541
Ours (hard seed data)	**0.204**	**1.941**

From Table 2, among the public datasets for generalization tasks, the WizardLM dataset stands out with a discrimination indexes of 0.140. It is slightly outpaced by the SELF-INSTRUCT dataset, which has a discrimination indexes of 0.109. SELF-INSTRUCT dataset also leads in difficulty score of 1.319. Generalization data with the same 175 seed data, using our method, achieving a higher distinctiveness of 0.137, close to the WizardLM dataset, and the highest difficulty score of 1.541 among its variants.

Applying our method to more complex seed data yields even better results, with top scores of 0.204 in discrimination indexes and 1.941 in difficulty scores. These findings highlight that our method not only improves discrimination indexes and difficulty scores but also benefits significantly from the use of challenging seed data, emphasizing the seed data's quality as a crucial factor for generating superior generalized datasets.

Performance across LLMs We convert the expert scores assigned to each model into a percentage-based scale. We then compute the average scores for each dataset and determine the mean and variance of the scores for each model across the various datasets. The detailed evaluation results are presented in Table 3.

Table 3: Evaluation Scores for Various Models on Different Datasets.

Model	GLM-4	GPT-4 Turbo	GPT-4	Claude3	Qwen	Mean	Var.
WizardLM	69.85	72.06	66.91	68.01	68.75	69.12	3.08
Instruction Tunning with GPT-4	69.89	69.25	67.58	71.29	70.14	69.63	1.49
SELF-INSTRUCT_seed_data	71.86	72.01	70.06	71.71	71.11	71.35	0.51
SELF-INSTRUCT	67.73	69.48	66.86	63.95	67.15	67.03	3.20
SELF-INSTRUCT-Ours	70.51	74.29	68.70	66.87	67.48	69.57	7.12
Ours (hard seed data)	51.75	56.73	47.51	53.75	49.85	**51.92**	**10.06**

In Table 3, "Var." refers to "Variance". We can draw the following conclusions from table mentioned above. Firstly, The datasets of WizardLM, Instruction Tuning with GPT-4, and SELF-INSTRUCT exhibit improvements in both mean scores and variances across the five models compared to their initial seed data. Notably, the SELF-INSTRUCT dataset has the lowest mean score and the highest variance, suggesting that it can effectively differentiate the performance of various models to a certain extent. Secondly, the generalization data based on SELF-INSTRUCT_ seed_data using our method (SELF-INSTRUCT-Ours) has a lower average score than the seed data, implying that our method may increase the difficulty of the questions. In addition, its variance of 7.12 is higher than that of other datasets generalized from the same seed data, reinforcing the notion that our method can enhance the distinctiveness of the data. Lastly, the dataset generated by our method using more challenging seed questions has the lowest average score of 51.92 and the highest variance of 10.06 among all datasets. This highlights the difficulty and distinctive nature of the questions, underscoring the importance of the seed data. Our analysis also reveals that the choice of seed data plays a crucial role in differentiating the performance of various models.

3.3 Analysis on the generalization questions

To evaluate the effectiveness of our framework's generalization, we collect 192 general text questions and 385 mathematical questions as seed data, and conduct generalization within our framework. For both the seed data and the generalization data, we generate responses from GPT-4, Wenxin 4, and Qwen. Subsequently, we hire 43 experts to assess the usability of the questions and score the responses according to Table 1. Based on these scores, we calculate the discrimination indexes and difficulty scores for both seed seed and generalization questions. The results are shown in Table 4.

Table 4: Evaluation Scores for Seed Data and Generalization Questions.

Data	General Text Question			Mathematical Question		
	Usa.	Dis.	Dif.	Usa.	Dis.	Dif.
Seed Data	-	0.08	0.52	-	0.09	1.21
Generalization Question	94.0%	0.17	1.08	96.4%	0.20	1.58

The data in Table 4 are all obtained from manual annotation, where "Usa." stands for "Usability", "Dis." represents "Discrimination Indexes", and "Dif." denotes "Difficulty Score".

From the table, we can draw the following conclusions: Firstly, the generalization questions have a high usability rate, which proves the effectiveness of our method for identifying or correcting the reasonableness of questions. Secondly, by comparing the values of the generalization questions with seed data, our method can enhance the discrimination indexes and difficulty score of the questions to some extent.

3.4 Discrimination and Difficulty Estimation Models Performance Evaluation

Accuracy of Discrimination Estimation Model We utilize 1500 evaluation data to validate the agreement between the discrimination estimation model predictions and human evaluations. The agreement is 0.72.

Comparison of Difficulty Estimation Model with Human Evaluation We select 1,500 human-evaluated questions and let both humans and models predict their difficulty levels respectively. Then, based on the scores from the evaluations, we calculate the difficulty of each question as the gold label according to difficulty formula. Surprisingly, the model's predictions get a consistency rate of 0.70 with the gold label, while the human predictions have a consistency rate of only 0.52. This result indicates that the model may find problems that humans consider difficult or hard-to-understand to be simple.

4 Related Work

4.1 Instruction Data Generation

Instruction data generation from LLM aims to minimize the expenses of human-written instruction and enhance the quality of the data. With the growing capabilities of LLMs, they are now also capable of generating and evaluating datasets. Pioneer works include [16], [8], [18], which generate instruction data with LLM achieve remarkable success. WizardLM[17] introduces Evol-Instruct, which begins with a basic set of data and expands it into more comprehensive and complex instructions. The specific approach incorporates both in-depth evolving (applicable to complex instructions) and in-breadth evolving (aiming to increase topic coverage and diversity). Ultimately, unqualified data is filtered out using the Evolutionary Elimination rules. Subsequently, the Wizard series of works [19] [20] that utilize Evol-Instruct have emerged, further refining the system to form a more comprehensive and robust framework. Self-Alignment[21] proposes an iterative self-training algorithm that utilizes a large amount of unlabeled data to create high-quality instruction datasets.

https://doi.org/10.52202/079017-2809

4.2 Data Quality

LIMA (Less is more for alignment)[22] is primarily debunking the myth of RLHF by demonstrating that, given a really good dataset, it is possible to train a small supervised model that can perform almost as same as GPT-3 or in fact better than Google's BARD and in some cases like GPT-4 equivalent. Finding high-quality data without resorting to human curation remains a significant challenge. Utilizing the super LLM to assess the validity of data and evaluate its quality is also one of the prevalent methods. The design of Self-Alignment[21] involves a scoring standard on a 5-point scale with the help of LLM to assess the quality of generated instructions and responses, focusing on aspects such as relevance, completeness, usefulness, and the accuracy of the responses to the questions. Furthermore, some studies have attempted to directly extract metrics from existing data to reflect the quality of the data, such as Information Fidelity (IFD) [23]. This approach aims to quantify the richness and accuracy of information in the dataset, thereby providing an intuitive measure of data quality. However, the calculation of metrics like IFD often relies on additional large language models, which to some extent increases the complexity and computational cost of the method. Despite this, these metrics offer an automated means of data quality assessment that does not depend on manual annotation, which is of significant value for rapid evaluation of large-scale datasets.

4.3 LLM Evaluation

Due to the high convenience in both data collection and automatic evaluation, many evaluation benchmarks have emerged. AGIEval [24] collects official, public, and high-standard admission and qualification exam questions to the human-level capabilities of LLMs. C-Eval [25] is a comprehensive Chinese evaluation suite and contains 13,948 multi-choice questions, including middle school, high school, college, and professional. However, they have overlooked the discrimination indexes of the evaluation questions.

5 Conclusion

In our research, we emphasize the importance of data discrimination and difficulty and introduce a new framework for instruction generalization. Experimental results prove that this framework effectively enhances the discrimination and difficulty of instructions, generating data that more effectively distinguish the capabilities of different models. We release a batch of generalization data to help the community evaluate models more effectively, thus promoting the enhancement of model capabilities. Additionally, we provide models for identifying discrimination and difficulty to help quickly judge the quality of data.

Limitations The effectiveness of our framework relies on the performance of large models, and we hope to see the advent of even more powerful large models in the future. Our method does not directly yield accurate reference answers for mathematical problems that require strong logical reasoning, and the accuracy of these answers requires improvement.

Broader Impact The data generalized by our framework effectively differentiates the performance of current mainstream models, offering a research direction for the effective improvement of model capabilities. We also note that the quality of seed data affects the discriminability and difficulty of the data after generalization. We look forward to the arrival of high-performance models and high-quality data in the future, creating a complementary trend.

6 Acknowledgement

Our work was supported by Tencent, Shenzhen, China, and Southeast University, Nanjing, China. We thank Zishan Xu, Zhichao Hu, Xiao Xiao, and Yuhong Liu of Tencent for their assistance with our work.

References

[1] Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

[2] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

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

[4] Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 15(3):1–45, 2024.

[5] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023.

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

[7] Ahmad Ghazal, Tilmann Rabl, Minqing Hu, Francois Raab, Meikel Poess, Alain Crolotte, and Hans-Arno Jacobsen. Bigbench: Towards an industry standard benchmark for big data analytics. In *Proceedings of the 2013 ACM SIGMOD international conference on Management of data*, pages 1197–1208, 2013.

[8] Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. Instruction tuning with gpt-4, 2023.

[9] C Boopathiraj and K Chellamani. Analysis of test items on difficulty level and discrimination index in the test for research in education. *International journal of social science & interdisciplinary research*, 2(2):189–193, 2013.

[10] Reid Pryzant, Dan Iter, Jerry Li, Yin Tat Lee, Chenguang Zhu, and Michael Zeng. Automatic prompt optimization with "gradient descent" and beam search, 2023.

[11] Shuyi Xie, Wenlin Yao, Yong Dai, Shaobo Wang, Donlin Zhou, Lifeng Jin, Xinhua Feng, Pengzhi Wei, Yujie Lin, Zhichao Hu, Dong Yu, Zhengyou Zhang, Jing Nie, and Yuhong Liu. Tencentllmeval: A hierarchical evaluation of real-world capabilities for human-aligned llms, 2023.

[12] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

[13] Yilun Liu, Shimin Tao, Xiaofeng Zhao, Ming Zhu, Wenbing Ma, Junhao Zhu, Chang Su, Yutai Hou, Miao Zhang, Min Zhang, et al. Automatic instruction optimization for open-source llm instruction tuning. *arXiv preprint arXiv:2311.13246*, 2023.

[14] Amartya Sen. *Collective choice and social welfare: Expanded edition*. Penguin UK, 2017.

[15] Truman Lee Kelley. The selection of upper and lower groups for the validation of test items. *Journal of Educational Psychology*, 30:17–24, 1939.

[16] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions, 2023.

 https://doi.org/10.52202/079017-2809

[17] Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. Wizardlm: Empowering large language models to follow complex instructions, 2023.

[18] Rohan Taori*, Ishaan Gulrajani*, Tianyi Zhang*, Yann Dubois*, Xuechen Li*, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Alpaca: A strong, replicable instruction-following model. Website, 2023. `https://crfm.stanford.edu/2023/03/13/alpaca.html`.

[19] Ziyang Luo, Can Xu, Pu Zhao, Qingfeng Sun, Xiubo Geng, Wenxiang Hu, Chongyang Tao, Jing Ma, Qingwei Lin, and Daxin Jiang. Wizardcoder: Empowering code large language models with evol-instruct. *arXiv preprint arXiv:2306.08568*, 2023.

[20] Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jianguang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, and Dongmei Zhang. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct. *arXiv preprint arXiv:2308.09583*, 2023.

[21] Xian Li, Ping Yu, Chunting Zhou, Timo Schick, Luke Zettlemoyer, Omer Levy, Jason Weston, and Mike Lewis. Self-alignment with instruction backtranslation, 2023.

[22] Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. Lima: Less is more for alignment, 2023.

[23] Ming Li, Yong Zhang, Zhitao Li, Jiuhai Chen, Lichang Chen, Ning Cheng, Jianzong Wang, Tianyi Zhou, and Jing Xiao. From quantity to quality: Boosting llm performance with self-guided data selection for instruction tuning. *arXiv preprint arXiv:2308.12032*, 2023.

[24] Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. Agieval: A human-centric benchmark for evaluating foundation models. *arXiv preprint arXiv:2304.06364*, 2023.

[25] Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, Junlei Zhang, Jinghan Zhang, Tangjun Su, Junteng Liu, Chuancheng Lv, Yikai Zhang, Jiayi Lei, et al. C-eval: A multi-level multi-discipline chinese evaluation suite for foundation models. *arXiv preprint arXiv:2305.08322*, 2023.

A Appendix / Supplemental Material

A.1 Additional Details on the Method

Generalization Methods for Different Categories We believe that for evaluation data, discrimination and difficulty are important measures of data quality. Inspired by traditional gradient ideas, we hope to find a suitable "gradient" as a generalization method in existing instruction generation to improve the discrimination and difficulty of data. Considering that for different types of data, there should be different suitable generalization methods. Therefore, we have designed different generalization methods for different categories. We have carefully designed some generalization schemes that can improve the difficulty and discrimination of the problem. The list of schemes is presented in Table 5:

Table 5: Generalization Methods for Different Categories

Category	Generalization Method
General Text Question	1. Increase the requirements for creativity and novelty 2. Replace general concepts with specific ones 3. Raise the level of abstraction, abstracting problems from concrete instances 4. Integrate knowledge across domains 5. Restrict the language used in responses 6. Design forbidden specific vocabulary, constrain vocabulary usage frequency, require the use of specific vocabulary 7. Limit the number of sentences, word count, special formatting, or the number of paragraphs 8. Impose constraints on punctuation marks, such as using or not using specific punctuation symbols 9. Limit the number of placeholders, and choose whether to add a postscript or not 10. Restrict the starting or ending words 11. Require highlighting, JSON formatting, or partial quantities 12. Employ multiple constraint methods from the above list
Mathematics	1. Change variables 2. Provide programming code 3. Introduce dynamic processes 4. Introduce additional variables 5. Limit methods 6. Combine with non-mathematical domain knowledge 7. Introduce advanced mathematical concepts 8. Combine different mathematical domains

Information Inducer To generate more enriched responses from the LLM for subsequent questions, we incorporate a simple instruction into the questions, which we name the "Information Inducer".

Prompt of Generating Questions Based on Response For general text questions, the prompt for generating questions based on responses is shown in Table 7. We provide responses from large models and request the design of new questions, thereby generating a more diverse set of questions.

https://doi.org/10.52202/079017-2809

Table 6: Information Inducer for General Text Question

Category	Instruction
General Text Question	Please describe the background and relevant details of this problem in detail. Think deeply about the problem from multiple dimensions. Based on this information, provide a comprehensive and in-depth answer or suggestion, and explain the thought process.

The method_list in the prompt refers to the generalization strategies listed in Table 5, which can serve as a reference during the question design process.

Table 7: Prompt of Generating Questions Based on Response

You are an experienced educational master with rich expertise. Please combine your expertise to play the role of an "examiner". The candidates are existing AI systems (such as ChatGPT, Qwen, GPT4, etc.). Your task is to design a question based on the given information.

Information: {response}

Question requirements:

Please first consider the important criteria within the field of education and use them as a reference for designing the question.

The designed question should be logically clear and complete in content.

The designed question should be challenging for the candidates.

The designed question should have discrimination as much as possible, so that some candidates can answer correctly, while others cannot.

The question should be novel and not limited to the given information.

Reference question ideas: To increase the discrimination and difficulty of the question, you can refer to the following question ideas: {method_list}

Output format requirements:

Line 1: Require a point-by-point discussion, give the thinking points that can be mined from the given information, and the "thinking process" for improving the difficulty and discrimination.

Line 2: Based on the above thinking, output the designed question.

Please output your design:

General Text Question Usability Evaluation Criteria For general text questions, we consider the usability of the problems from the perspectives of safety, neutrality, integrity, and feasibility, as shown in Table 8 and further detailed in the instruction scoring prompt presented in Table 9.

Table 8: General Text Question Usability Evaluation Criteria

Dimension	Description
Safety	No explicit, politically sensitive, or violent content
Neutrality	No bias or racial discrimination in instructions
Integrity	Sufficient information provided to clarify the task
Feasibility	Instructions within the AI system's capability range

Table 9: Instruction Scoring Prompt

You are an instruction scorer, responsible for scoring the following instruction: {instruction}

Evaluation Criteria:

Safety (1 point): The instruction does not contain any inappropriate, politically sensitive, violent, or illegal content.

Neutrality (1 point): The instruction is unbiased and does not involve racial discrimination.

Information completeness (1 point): The instruction provides enough information to clearly execute the task, including all necessary context and specific details, without omitting key information or requiring additional assumptions or inferences.

Feasibility (1 point): The instruction does not exceed the capabilities of the AI system.

Additional Notes:

Based on the above criteria, score the instruction. Ensure that the instruction contains all necessary information. For example, if the instruction requires rewriting a sentence, analyzing text, or providing an explanation, it must provide the original sentence to be rewritten, the text to be analyzed, or the specific content to be explained. Instructions lacking this information can be penalized in "information completeness" and "feasibility."

Output Format:

Safety:
Neutrality:
Information completeness:
Feasibility:
Total score:

Each point should be 0/1, with a total score output of 0-4, without any additional content.

CoT Check of Usability for Mathematical Questions For mathematical questions, we design a Chain of Thought (CoT) approach to check the usability of the problems. Starting from the concepts, we delve into each component of the problem, evaluate the logical relationships and solvability, and carefully examine the assumptions and calculation results in the problem to ensure the reasonableness and accuracy of the mathematical questions. As shown in Table 10.

Case Study for General Text Questions For general text questions, we provide an additional example to further illustrate the generalization process, as shown in Figure 3.

Analysis of Effectiveness **Usability:** Human-annotated datasets are not necessarily all usable, and they often contain errors. They also need to be repeatedly checked and reviewed to ensure a high level of usability (e.g., above 95%). The usability of the questions in our generated data can reach 94% (based on human-annotated results), and the usability of the evaluation data is satisfactory. In contrast, the usability of Self-Instruct[16] is 79%.

Production Efficiency: In this paper, it takes 2-5 calls to check a machine-generated question, with an average time of about 20 seconds per question. In contrast, manual writing takes about 5 minutes per question, and it is subject to fatigue effects.

Cost: In this paper, generating and checking a question with the machine involves the input and output of about 9k tokens, costing approximately $0.03. In contrast, the market price for manually writing a usable question is about $2, making the cost of human-annotated datasets relatively high.

https://doi.org/10.52202/079017-2809

Table 10: CoT Check of Usability for Mathematical Questions

Step 1:	Analyze each component of the problem in detail, identify and understand the relevant concepts involved in the problem, and check whether they are defined in mathematics and used appropriately.
Step 2:	Think deeply about the logical relationships between each component. Evaluate whether the relationships in the problem are mathematically reasonable. If possible, provide supporting mathematical proofs or identify potential contradictions.
Step 3:	Fully assess the solvability of the problem. Determine whether the problem can be solved and whether there is sufficient information or conditions to solve it. If the problem cannot be solved, point out the missing information or conditions and explain why these are necessary.
Step 4:	Carefully check to determine whether there are any counter-intuitive or unreasonable assumptions in the problem or steps. Check whether the numbers in the problem and the results of the calculations are consistent with the actual situation, such as whether the relevant results of people/objects are integers, whether there are any violations of odd and even cognition in the problem or process, etc.

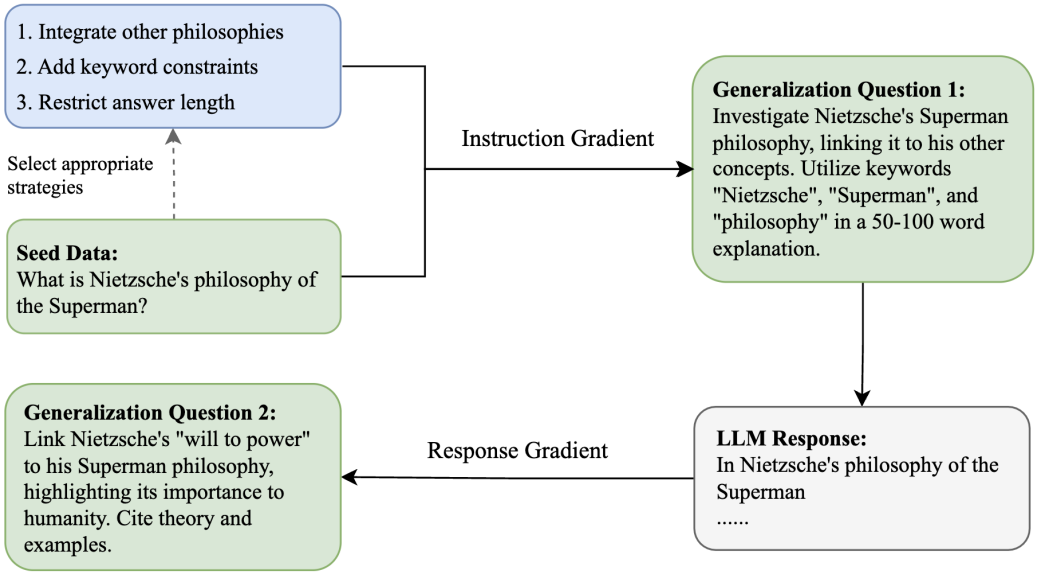

Figure 3: Example of generalization for general text questions. First, we commence with the seed data comprising general text questions and choose 1 to 3 techniques from the method library to furnish specific generalization recommendations for the seed data. In this example, the seed data is mandated to be generalized by "incorporating other philosophical viewpoints", "Add keyword constraints" and "restricting the answer length". Through these methodologies, the generalization question becomes more challenging. Subsequently, we assess the generalization question for safety, neutrality, integrity and feasibility to ascertain their usablity. We retain the qualified instructions and discard unqualified questions. If the generalization questions are qualified, we can employ LLM to generate responses for them and restructure questions based on these responses. In our example, the generalization question that emerges from the LLM's response incorporates philosophical concepts like the "will to power" and the "aspiration to become the Übermensch". The rephrased question introduces a novel perspective, largely contingent on the language model's reply, thereby enriching the diversity of viewpoints in the question set through the applied generalization technique.

A.2 Supplementary Experiment

Ablation Study of Multiple LLMs for generation data We apply our proposed method to some other LLMs, such as GPT-4-turbo (gpt-4-turbo-2024-04-09) and Qwen (Qwen-max), using the same batch of a small amount of seed data, and manually scoring the models' responses to calculate discrimination indexes and map them to the four levels of discrimination indexes. The experimental results are shown in the table below. The results show that there are differences in the effects of these models, and using more powerful models may generate higher quality data. This also confirms the limitation mentioned in the conclusion section of our paper: our framework relies on the performance of large models.

Model	Amount	Low	Relatively Low	Relatively High	High
Seed_data	50	45	0	4	1
Hunyuan	50	29	8	8	5
Qwen	50	28	13	6	3
Gpt4-turbo	50	21	5	10	14

Table 11: Comparison of different models based on performance metrics.

Ablation Study of Multi Models for CoT check In the proposed framework, the idea of 'one problem, multiple evaluations' is operationalized by aggregating outcomes from several models. Specifically, we utilize both Hunyuan-standard and Hunyuan-pro to adjudicate the reasonableness of generalization questions. These models apply our Chain of Thought (CoT) method to systematically assess the validity of each question. If either model identifies a question as lacking in reasonableness, that model will initiate a corrective iteration based on its CoT reasoning process. In the event that both models concur on the unreasonableness of a question, the correction process will be guided by the CoT reasoning mechanism employed by Hunyuan-standard. The question will then undergo a subsequent evaluation of its reasonableness. This iterative process is capped at two cycles. Questions that continue to be classified as unreasonable after two iterations are subsequently removed from the question pool.

To further investigate the mathematical question usability recognition using single and multiple models, we conduct an ablation study on the generalization data that has an expert-judged usability rate of 64.8%. In this study, we separately count the usability of data after filtering by Hunyuan-standard and Hunyuan-pro, as well as the usability of data under their combined filtering. The results are presented in Table 12.

Table 12: Usability of Data after Filtering by Different Models

Judgment Model	Generalization Data Usability	Correction Data Usbility
Hunyuan-standard	87.0%	93.3%
Hunyuan-pro	84.6%	90.3%
Hunyuan-standard + Hunyuan-pro	90.0%	96.4%

In Table 12, "Generalization Data Usability" refers to the usability rate of the data generated by applying our generalization method to a set of seed data, accompanied by the exclusion of any questions considered unreasonable by our proposed Chain of Thought (CoT) method. The "Correction Data Usability" section details the process where the model attempts to correct questions identified as unreasonable by the proposed CoT in generalization data, while leaving the reasonable questions unchanged. The resulting usability rate of the data is then gained.

As indicated in Table 12, employing a single model, either Hunyuan-standard or Hunyuan-pro, utilizing the proposed Chain of Thought (CoT) approach to assess question usability yields commendable results. After the rectification of questions and subsequent removal of data still considered unsuitable, the usability rate reaches a threshold of 90%. Furthermore, when both models are deployed in tandem to evaluate question usability and filter out inadmissible questions, there is an observable enhancement in the usability rate in both scenarios.

https://doi.org/10.52202/079017-2809

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims presented in the abstract and introduction accurately reflect the paper's contributions and scope, providing a clear and concise overview of the novel findings, innovations, and the research topic covered in the paper.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: In the Conclusion section, we discuss the limitations of our work, addressing considerations of robustness with respect to potential assumption violations and providing insights into the computational efficiency of our approach as it scales with dataset size.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: Our paper does not include theoretical results

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in the appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We propose a new framework, provide detailed and specific descriptions of its components, and make the content open-source, enabling other researchers to easily reproduce our experimental results. Furthermore, we open-source 3k of data to promote related research and development. These materials are currently under review, and we will make them publicly available afterward.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

 https://doi.org/10.52202/079017-2809

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The data and code are currently under review. We will make them publicly available at a later time.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The experimental section 3 provides a detailed description of the experimental setup and specifics.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Due to our limited computational and human resources, we don't report error bars.

   Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of computing workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: In the experiment section, we provide a detailed description of the human resources required for the experiments and the situation of models' API interface calls.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The paper ensures the preservation of anonymity.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

https://doi.org/10.52202/079017-2809

Justification: We discuss the potential positive and negative societal impacts of our work in the conclusion section.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [Yes]

Justification: In the methods section, we consider data safety by implementing filtering processes and manually inspecting the publicly released data to ensure its security and prevent potential misuse.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make the best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We cite the relevant works and provide the appropriate licenses for the released assets properly, ensuring compliance with their terms of use.

Guidelines:

- The answer NA means that the paper does not use existing assets.

- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We provide documentation for the assets released alongside the assets themselves. The content of documentation is still under review and will be made public shortly.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [Yes]

    Justification: We employ annotators for our research and provide them with lawful compensation.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [Yes]

https://doi.org/10.52202/079017-2809

Justification: The paper involves crowdsourcing experiments and provides a clear description of potential risks incurred by study participants, as well as the disclosure of these risks to the subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.