Error-driven Data-efficient Large Multimodal Model Tuning

Barry Menglong Yao
UC Davis
bmyao@ucdavis.edu

Qifan Wang
Meta AI
wqfcr@meta.com

Lifu Huang
UC Davis
lfuhuang@ucdavis.edu

Abstract

Large Multimodal Models (LMMs) have demonstrated impressive performance across numerous academic benchmarks. However, fine-tuning still remains essential to achieve satisfactory performance on downstream tasks, while the task-specific tuning samples are usually not readily available or expensive and time-consuming to obtain. To address this, we propose an error-driven data-efficient tuning framework that aims to efficiently adapt generic LMMs to newly emerging tasks without requiring extensive task-specific training samples. In our approach, a generic LMM, acting as a student model, is first evaluated on a small validation set of the target task, and then a more powerful model, acting as a teacher model, identifies the erroneous steps within the student model's reasoning steps and analyzes its capability gaps from fully addressing the target task. Based on these gaps, targeted training samples are further retrieved from existing task-agnostic datasets to tune the student model and tailor it to the target task. We perform extensive experiments across three different training data scales and seven tasks, demonstrating that our training paradigm significantly and efficiently improves LMM's performance on downstream tasks, achieving an average performance boost of 7.01%[1].

1 Introduction

Pretrained large multimodal models (LMMs), such as GPT-4 (Achiam et al., 2023) and LLaVA (Liu et al., 2024a), have demonstrated strong performance across various academic benchmark datasets (Xu et al., 2022; Reddy et al., 2022; Liu et al., 2024c; Lu et al., 2022; Yue et al., 2024; Yu et al., 2023). However, when leveraging LMMs for real-world applications, despite direct task adaptation with techniques such as prompting (Radford et al., 2019; Wei et al., 2023; Qi et al., 2023; Yao et al., 2024) or in-context learning (Brown, 2020; Jiang et al., 2024; Zhao et al., 2024b; Doveh et al., 2024), careful fine-tuning on a substantial amount of task-specific training samples is still essential in order to achieve satisfactory performance (Luo et al., 2022; Gu et al., 2021; Liang et al., 2023; Yao et al., 2023), while such task-specific training samples are usually not readily available or expensive and time-consuming to achieve. Therefore, a critical question that we would like to answer is: *How can we effectively tune large multimodal models for newly emerging problems without requiring a large amount of task-specific training samples?*

One potential solution is to apply data augmentation methods to automatically synthesize or enlarge the training samples (Lee et al., 2024b; Dai et al., 2023; Li et al., 2024b; Zhao et al., 2024a; Nayak et al., 2024; Xu et al., 2023b). However, they usually lead to undesired effects, such as introducing significant *bias* into the downstream tasks (Angelakis and Rass, 2024; Lin et al., 2024; Muthukumar et al., 2020; Hastie et al., 2022) or causing *model collapse* (Shumailov et al., 2023; Feng et al., 2024), where models tuned from synthesized training samples tend to forget the true underlying distribution of human-generated datasets. Additionally, several recent studies explored selecting relevant tasks or data samples from external resources to fine-tune the models for target tasks, where the selection is based on the similarity between the evaluation instances of the target task and training samples of other tasks using either features such as n-grams and task instructions (Lee et al., 2024a; Xie et al., 2023; Gururangan et al., 2020) or gradients calculated from the model (Xia et al., 2024a; Han et al., 2023). However, these approaches either necessitate a high degree of alignment between the surface forms of external datasets and the target task or rely on backward passes that are computationally intensive due to the large size of the external datasets.

[1] The programs are publicly available at https://github.com/PLUM-Lab/DELAMO_LMM_Tuning.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 20289–20306
July 27 - August 1, 2025

In this work, we propose a novel *error-driven*, *data-efficient* tuning paradigm that enables the effective adaptation of generic, pre-trained large multimodal models (LMMs) to diverse and emerging downstream tasks, while minimizing the need for extensive task-specific training samples. This paradigm is motivated by the *gap detection and filling* process in human learning (Bambrick-Santoyo, 2010), where learners identify knowledge gaps and incrementally fill them through targeted exploration. Based on this motivation, we design a teacher-student framework where a pre-trained LMM, acting as the student model, is first applied to a small set of validation samples specific to the target task. The student model's predictions are then analyzed, and based on its errors, a teacher model—typically another large multimodal model (e.g., GPT-4o-mini)—is designed to identify the erroneous steps within the student model's reasoning processes, and further analyze and summarize its missing skills, representing the capability gaps preventing the student model from fully addressing the target task. After identifying these gaps, a set of tuning samples that are specifically related to the missing skills is retrieved from existing task-agnostic, large-scale supporting datasets, to fine-tune the student model.

To evaluate the effectiveness of our framework, we employ different student models, including LLaVA-7B (Liu et al., 2024a) and Qwen2-VL-7B (Wang et al., 2024), and teacher models, including GPT-4o-mini (Achiam et al., 2023) and LLaVA-OneVision-72B (Li et al., 2024a), and conduct extensive experiments across seven tasks and datasets, including MM-Bench (Liu et al., 2024c), a comprehensive benchmark covering a wide range of multimodal processing tasks, and six downstream tasks including ScienceQA (Lu et al., 2022), Appliance Classification (Lin et al., 2014), Furniture Classification (Lin et al., 2014), Living Thing Classification (Li et al., 2022), Vision Question Answering (Zhu et al., 2016), and Image Caption Match (Lin et al., 2014). We utilize Vision-Flan (Xu et al., 2024) as the external supporting dataset as it covers hundreds of existing human-labeled tasks and datasets. Across different numbers of tuning samples retrieved from the supporting dataset, our approach significantly outperforms other data selection baselines as well as the LMM that is fine-tuned on the whole supporting dataset, highlighting the efficiency and effectiveness of our error-driven, data-efficient tuning framework in task adaptation.

Our contributions are summarized as follows:

- We propose a novel error-driven, data-efficient tuning framework that identifies capability gaps in LMMs and retrieves targeted tuning samples from existing datasets to effectively adapt them to new downstream tasks without requiring extensive task-specific training samples.
- We conduct comprehensive experiments, demonstrating that our framework significantly surpasses all baseline methods in effectively adapting generic LMMs to specific downstream tasks while incurring minimal training costs.

2 Related Work

Error-driven Learning Inspired by cognitive science, error-driven learning (Carpenter and Grossberg, 1987; Hoppe et al., 2022) enhances model performance by updating parameters based on error samples (Rumelhart et al., 1986) or explicitly analyzing and addressing errors. For instance, Yang et al. (2023) and Wang and Li (2023) directly prompt large language models (LLMs) to summarize error-driven guidance and integrate it into subsequent prompts. Akyürek et al. (2023) and Xu et al. (2023a) introduce critique generators to refine predictions during inference. Other studies (Lee et al., 2024b; An et al., 2023; Li et al., 2023b; Chen et al., 2023a; Wang and Huang, 2024) propose targeted data augmentation that automatically generates synthetic data using error samples. Unlike these methods, our approach fine-tunes LMMs by retrieving training samples from large-scale, domain-agnostic datasets, addressing missing skills identified from error samples.

Data Selection Data selection is often framed as a coreset selection problem (Phillips, 2016), aiming to identify a subset of training examples that achieves comparable performance to the full dataset. This is typically done by assessing training data quality (Liu et al., 2024d; Chen et al., 2023b; Zhou et al., 2024; Toneva et al., 2018; Sener and Savarese, 2017; Killamsetty et al., 2021; Xia et al., 2024b) or selecting high-uncertainty samples (Kung et al., 2023; Liu et al., 2024b). Targeted data selection refines this approach by choosing fine-tuning data aligned with the target distribution, using similarity measures based on surface features (Lee et al., 2024a; Xie et al., 2023; Gururangan et al., 2020) or LLM gradient vectors (Xia et al.,

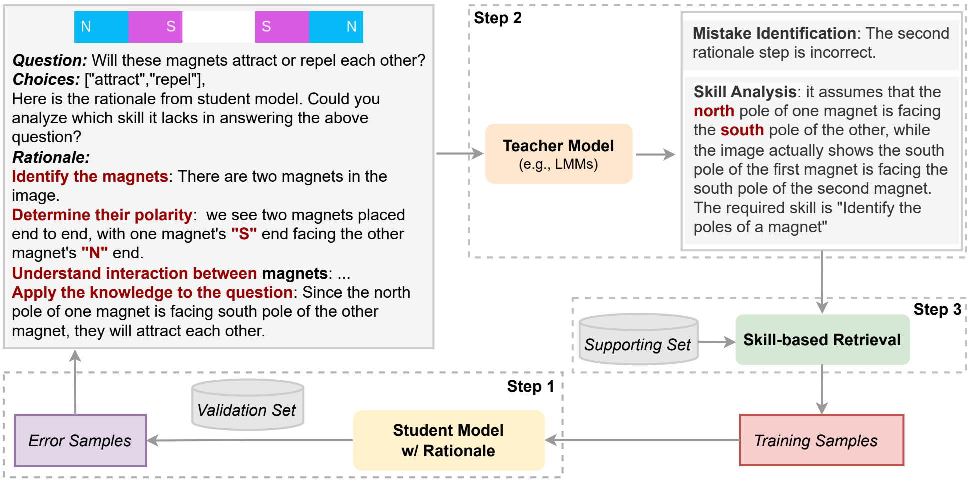


Figure 1: Overview of the error-driven data-efficient tuning paradigm.

2024a; Han et al., 2023). Unlike these methods, our approach directly identifies LMM weaknesses from error samples, enabling more data-efficient and computation-efficient sample selection.

Curriculum Learning Curriculum Learning (CL) trains models in a structured order, progressing from easy to hard samples. Early CL studies (Bengio et al., 2009; Spitkovsky et al., 2010) relied on rule-based criteria (e.g., training on shorter sequences first). Self-paced learning methods (Kumar et al., 2010; Lee and Grauman, 2011; Ma et al., 2017) dynamically select samples based on model performance, training loss, or likelihood. Recent teacher-student approaches (Matiisen et al., 2017; Kim and Choi, 2018; Hacohen and Weinshall, 2019; Zhang et al., 2019) use reinforcement learning to guide selection. Our method extends this framework by introducing Mistake Identification and Skill Analysis to efficiently detect and address the student model's weaknesses.

3 Approach

3.1 Overview

Given a new task with a test set $\mathcal{D}_{test}$ and a validation set $\mathcal{D}_{val}$, we aim to efficiently adapt a generic, pre-trained large multimodal model (LMM) to it without requiring extensive task-specific training samples. To achieve this, we propose an error-driven data-efficient tuning framework, as shown in Figure 1, consisting of three iterative steps: Step 1 (**Error Collection**) identifies error samples by evaluating the student model's predictions and rationales on validation samples. Step 2 (**Mistake Identification and Skill Analysis**) uses a teacher model to pinpoint the key erroneous step and infer the missing skill needed for improvement. Note that while most downstream tasks require diverse skills, a pre-trained LLM may have already excelled in some, so we mainly focus on identifying and enhancing the missing skills in the given LMM. Step 3 (**Targeted Tuning**) further retrieves targeted samples from existing datasets to fine-tune the student model, refining its capabilities for the missing skills. These three steps iterate until the maximum number of iterations is reached. In the following, we provide details for each component.

3.2 Error Collection from Student Model

Given a target task with a validation set $\mathcal{D}_{val}$, we leverage a generic and pre-trained LMM as the student model $\mathcal{M}_S$, which is prompted to generate a sequence of intermediate reasoning steps (Wei et al., 2023) and a final answer for each validation sample. The intermediate reasoning steps are viewed as a rationale for the predicted answer. The LMM is prompted to specifically follow an answer format such as "The final answer is option A", and we will directly parse the final answer from the model's response based on the answer format.[2] An example prompt for ScienceQA task is shown in Figure 3 in Appendix B.1. We finally compare the predicted answer with gold answer for each validation example and obtain a set of error samples and their corresponding intermediate reasoning steps as rationales.

3.3 Mistake Identification

Given an error sample containing a question q, a wrong prediction y with a rationale r from the

[2]We also consider the variants of the answer format shown in Table 10 in Appendix A.

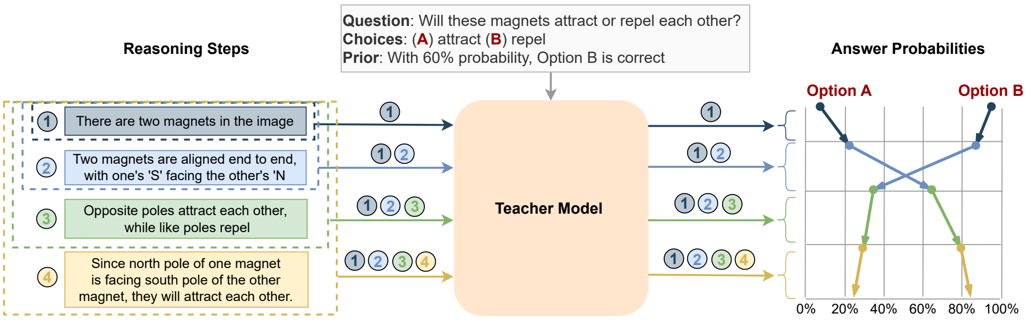


Figure 2: Example for illustrating the process of mistake identification. At each iteration, we append one more reasoning step into the prompt to ask the teacher model to answer the question and track the probability changes of all the candidate option tokens.

student model, and a gold answer $\tilde{y}$, we first split the rationale r into a sequence of reasoning steps $r = [r_1, r_2, ...]$. Here, we follow previous study (Tyen et al., 2024) and treat each sentence in the rationale as one reasoning step. The goal of **Mistake Identification** is to apply a new large multimodal model, e.g., GPT-4o-mini (Achiam et al., 2023), as a teacher model $\mathcal{M}_T$ to analyze the incorrect predictions from the student model, and locate the *Mistake Step* r_m, a.k.a., the most significant erroneous reasoning step that leads to the final incorrect answer, from the rationale. Motivated by previous studies (Tyen et al., 2024), we define the most significant erroneous reasoning step r_m as the first rationale step that leads to the prediction of the wrong answer y. For example, for the error sample shown in Figure 1, the second step "*one magnet's south end facing the other magnet's north end*" is identified as the mistake step as it contributes most to the final wrong prediction of the student model.

We propose an *answer-switch* based method to identify the mistake step, as shown in Figure 2. The core idea is to prompt the teacher model to respond to the same question using the rationales provided by the student model. We then analyze the changes in the probabilities of the candidate answers as each individual rationale step is incrementally appended. To encourage the teacher model to favor the correct answer at the beginning, we modify the prompt to include prior knowledge that indicates a higher probability for the correct answer, e.g., "*There is a probability of 60% that option B (repel) is correct*", and instruct the teacher model to rely on this prior knowledge if it lacks sufficient information to determine the answer. We then gradually append each reasoning step into the prompt of the teacher model $\mathcal{M}_T$ and monitor the changes in the model prediction, with the expectation that the probability of wrong answer y will gradually become higher after we append the erroneous reasoning steps.

Our preliminary experiments on ScienceQA with 100 error samples have shown that when the teacher model is provided with the correct answer as prior knowledge, it initially could assign a higher probability to the correct answer in 89% of cases. As the student model's incorrect reasoning steps are added, the teacher model shifts to the wrong answer in 70% of cases. Without prior knowledge, the correct answer receives a higher initial probability in 60% of cases, with a shift to the wrong answer in 43%. These results support the design of our *answer-switch* based approach for mistake identification. In addition, we also restrict the teacher model from accessing the image so that it's forced to choose the answer solely based on the reasoning steps of the student model.

Figure 4 in Appendix B.2 shows the prompt template for mistake identification. For each round of inference, the input prompt to $\mathcal{M}_T$ consists of the question "*Will these magnets attract or repel each other?*", the prior knowledge about the correct answer "*There is a probability of 60% that option B is correct*", and a subset of reasoning steps, while the output consists of a template-based answer, e.g., "*The answer is the option A*".[3] To determine the probability of each candidate option, we first identify the position of the option token (e.g., "A") in the answer, and obtain the probabilities of other candidate option tokens such as "B", "C", and "D", from the teacher model. This process is repeated as we sequentially append each reasoning step to the prompt, enabling us to track the probabilities of all answer options across iterations, e.g., $\{P(A|q, r_1), P(A|q, r_1, r_2), ..., P(A|q, r_1, ..., r_i)\}$, $\{P(B|q, r_1), P(B|q, r_1, r_2), ..., P(B|q, r_1, ..., r_i)\}$, respectively. Based on the change in probabilities of the correct answer "B" and the wrong answer

[3]For non-multiple-choice tasks, we convert them by setting the gold answer and the wrong prediction as two options.

"A", we identify the mistake step r_m as the first reasoning step that causes the probability of the wrong answer to be higher than the probability of the correct answer by a predefined margin δ and the margin is maintained for the following λ iterations:

$$m := \min\{i \mid \forall j \in \{0, \dots, \lambda - 1\},\\ P(A \mid q, r_1, \dots, r_{i+j}) - \delta \geq P(B \mid q, r_1, \dots, r_{i+j})\}$$

where δ is the probability gap between the wrong answer and the correct answer, and λ is the number of steps where the probability gap persists.[4]

3.4 Skill Analysis

After identifying the erroneous reasoning step r_m from the rationale of each error sample, we further perform **Skill Analysis**, where the teacher model $\mathcal{M}_T$ is prompted to summarize one missing skill s,[5] such as *identifying the poles of a magnet* in Figure 1, which is required to correct the wrong reasoning step r_m. Note that, for each error sample, we focus on one missing skill in one iteration and leave other missing skills for the following iterations. To achieve this goal, we design an in-context learning (ICL) (Wei et al., 2022a,b) based approach where the input of each in-context exemplar consists of a question together with its correct answer, complete rationale steps and a mistake step, and the output is the missing skill which is required to correct the mistake. The prompt template for **Skill Analysis** is shown in Figure 5 in Appendix B.3.

3.5 Targeted Tuning

After analyzing the missing skills for all the error samples from the validation set $\mathcal{D}_{val}$, we then retrieve a set of relevant training samples from a domain-agnostic large-scale supporting dataset [6] to construct a targeted tuning dataset $\mathcal{D}_{train}$ and utilize $\mathcal{D}_{train}$ to fine-tune the student model to enhance its capability and address the identified skill gaps for the target downstream task.

[4]We manually labeled the mistake step for 100 ScienceQA validation error examples and tuned δ and λ on them (see results and probability gap statistics in Appendix C and Section 4.5).

[5]We follow (Chen et al., 2023c) and define a skill as a unit of behavior with associated data X such that if the LMM is trained on dataset D, where $D \subseteq X$, it has improved performance on samples belonging to $X \backslash D$. See Appendix D for more details on skill definition.

[6]Our framework does not require the supporting dataset to be semantically similar to the downstream tasks. Instead, it capitalizes on underlying skills—such as counting and spatial relation recognition—that are shared between target tasks and existing task-agnostic instruction-tuning datasets.

Specifically, for each sample in the supporting dataset, we pre-compute a set of required skills by prompting the teacher model to follow in-context exemplars and provide detailed analysis of the skills that are required to achieve the correct answer. The prompt template is shown in Figure 6 in Appendix B.4. Then, for each error sample in $\mathcal{D}_{val}$, we apply BM25 (Robertson et al., 2009) to calculate similarity scores between its missing skill s and the concatenation of all required skills of each sample in the supporting dataset. The samples in the supporting dataset are then ranked according to the similarity scores, and the top-K samples are selected as the training samples to improve the missing skills of the student model.

4 Experiment

4.1 Experimental Setup

For evaluation, we experiment with two different student models, including the instruction-tuned LLaVA-v1.5-7B (Liu et al., 2024a)[7] and Qwen2-VL-7B (Wang et al., 2024; Bai et al., 2023)[8], and two different teacher models, including GPT-4o-mini (Achiam et al., 2023) (`gpt-4o-mini-2024-07-18`) and LLaVA-OneVision-72B (Li et al., 2024a)[9], and evaluate our framework on seven downstream tasks and datasets: **MM-Bench**, a generic benchmark dataset for evaluating large multimodal models and covering diverse categories of tasks such as *Attribute Recognition*, *Action Recognition*, *Object Localization*, and so on. MM-Bench is used to demonstrate the potential of our error-driven efficient-tuning framework as a post pre-training step to further improve the general capabilities of large multimodal models; and six downstream tasks, including **ScienceQA** (Lu et al., 2022), **Appliance Classification** (Lin et al., 2014), **Furniture Classification** (Lin et al., 2014), **Living Thing Classification** (Li et al., 2022), **Vision Question Answering** (Zhu et al., 2016), and **Image Caption Match** (Lin et al., 2014). For each of the downstream tasks, we sample 1K data points as the test set and 1K data points as the validation set. These tasks are employed to demonstrate the efficiency of our framework in adapting the generic pre-trained large multimodal model to spe-

[7]`https://huggingface.co/liuhaotian/llava-v1.5-7b`

[8]`https://huggingface.co/Qwen/Qwen2-VL-7B-Instruct`

[9]`https://huggingface.co/lmms-lab/llava-onevision-qwen2-72b-ov-chat`

Method	# of Tuning Samples	MM-Bench	Appliance Cls	Furniture Cls	Living Thing Cls	VQA	Image-Cap Match	ScienceQA
Pre-trained LMM	0	64.30	45.80	49.00	79.40	77.00	64.10	65.34
Random	10K	62.85	57.47	60.60	82.10	74.03	65.03	63.66
Superfiltering	10K	62.65	49.90	53.60	77.00	75.30	68.40	64.15
INSTA*	10K	63.25	60.00	64.10	89.20	72.20	74.70	62.52
Our Approach	10K	**63.86**	**62.10**	**64.80**	**90.60**	**76.00**	**77.70**	**65.89**
Random	30K	62.60	61.07	63.13	86.83	75.50	71.97	63.38
Superfiltering	30K	62.95	53.40	53.80	78.40	76.90	69.90	65.05
INSTA*	30K	63.25	61.90	66.10	92.90	72.10	76.90	65.39
Our Approach	30K	**64.01**	**62.20**	**67.10**	**93.30**	**77.30**	**80.00**	**67.53**
Random	100K	62.95	60.83	66.23	88.67	76.90	77.50	64.55
Superfiltering	100K	63.25	55.60	54.90	82.90	77.40	73.50	65.00
INSTA*	100K	62.05	62.90	66.80	92.80	74.00	77.60	65.25
Our Approach	100K	**64.41**	**64.10**	**67.70**	**93.60**	**79.00**	**80.10**	**68.02**
Full Data	1,552K	62.43	63.50	69.80	90.60	74.90	84.70	67.23
Validation Data	1K	63.86	59.90	57.80	89.00	77.40	67.80	65.39

Table 1: Evaluation results on seven downstream tasks with different numbers of tuning samples retrieved from the supporting dataset (%). **Full Data** means that the whole supporting dataset is used to tune the LMM while **Validation Data** stands for fine-tuning the pre-trained LMM on 1K validation samples of the target task.

cific downstream tasks. We use **Vision-Flan-1-million** (Xu et al., 2024)[10] as the supporting dataset as it covers hundreds of existing tasks and datasets created by humans.

We compare the student model tuned using our error-driven data-efficient tuning framework with four baselines: (1) **Pre-trained LMM**, which denotes the vanilla student model without any tuning; (2) **Random Sampling**, where the training samples are randomly sampled from the supporting set. This process was repeated three times, and the average performance is reported in Table 1. Detailed results for each run are reported in Table 11 in Appendix E. (3) **INSTA*** (Lee et al., 2024a), which ranks the training samples based on their SBERT (Reimers and Gurevych, 2019) similarity scores to the validation samples, and select the same number of samples for targeted tuning. and (4) **Superfiltering** (Li et al., 2024c), which utilizes a small GPT-2 model (Radford et al., 2019) to filter out the high-quality subset based on instruction-following difficulty (IFD) score (Li et al., 2023a). Additionally, to better demonstrate the effectiveness and efficiency of our error-driven model tuning framework, we also show the performance of the student model that is fine-tuned on the whole supporting dataset (**Full Data**) or the 1K task-specific validation samples (**Validation Data**).

4.2 Main Results

Table 1 shows the performance of our framework with LLaVA-v1.5-7B as the student model and GPT-4o-mini as the teacher model, using different numbers of tuning samples from the supporting dataset. We compare our approach with several baselines and can see that: (1) The pre-trained LMM underperforms on some tasks such as Appliance Classification (45.80% accuracy) and Furniture Classification (49.00% accuracy), highlighting the need for further fine-tuning; (2) Our error-driven tuning framework significantly improves performance across different training scales, with a notable 7.01% average boost across seven tasks at the 100K tuning sample scale compared to the pre-trained LMM; (3) By carefully analyzing the missing skills of the pre-trained LMM, our approach is consistently more effective at adapting it to the target task than other baselines across different training scales; (4) Random Sampling shows comparable performance as other baselines, which is consistent with previous studies (Xia et al., 2024b; Chen et al., 2024) and might be attributed to the positive effects of data diversity. However, it's unstable and sometimes results in poorer performance, e.g., 10K training scale for Image-caption Match; (5) Superfiltering performs the worst among baselines, as GPT-2's inability to process image input limits its performance; (6) Remarkably, using just 6% of the full supporting dataset (100K samples), our approach achieves at least 94.57% of the **Full Data** performance across all benchmarks and even outperforms the **Full Data** setting on five tasks, indicating that training LMMs with large-scale task-agnostic datasets may suffer from task interference issue (Wang et al., 2023; Shen et al., 2024) and hinder the development of task-specific capabilities, highlighting the necessity of targeted data selection for more efficient model adaptation; (7) More complex tasks usually require more training samples, e.g., Image-Caption-Match and Living Thing Classification can be significantly improved by our approach with 10K training samples while

[10]We removed all samples related to the seven evaluation tasks in Vision-Flan-1-million to ensure no overlap.

the VQA task requires 100K. (8) Based on the main results presented in Table 1 and the supplementary analysis in Table 12 (Appendix F), we observe that tasks involving fewer reasoning steps typically achieve greater performance improvements, whereas tasks with longer reasoning chains exhibit comparatively limited gains. We attribute this pattern to two primary factors: (a) Error Localization Complexity—the challenge of accurately identifying the erroneous reasoning step intensifies as reasoning chains grow longer; and (b) Inherent Task Difficulty—tasks requiring longer reasoning chains are inherently more complex, thus making them more challenging targets for performance enhancement.

Method	# of Validation Samples	# of Turning Samples	Furniture CIs	Image-Cap Match
Pre-trained	-	-	49.00	64.10
Our Approach	0.1K	10K	62.10	75.90
Our Approach	1K	10K	64.80	77.70
Our Approach	0.1K	100K	67.00	78.50
Our Approach	1K	100K	67.70	80.10

Table 2: Experiments with different sizes of validation set, with LLaVA-7B as the student model and GPT-4o-mini as the teacher model.

Requirement of a Small Validation Set While our error-driven, data-efficient tuning framework shows significant improvements on various downstream tasks, we acknowledge that the need for a validation set for each target task could limit generalizability. However, our approach only requires a small validation set—around 1K samples—which is more feasible than large, human-annotated, task-specific training datasets. To reduce this cost, we tested using a smaller validation set. As shown in Table 2, even with just 100 validation samples, our framework still enhances pre-trained LMMs, achieving a good cost-performance balance. However, performance slightly decreases compared to the 1K-sample setting, likely due to reduced diversity and skill coverage. Future work could explore using closed-source LMMs to generate pseudo-answers for an unlabeled validation set, reducing the need for manual labeling.

Results of Different Student and Teacher Models To demonstrate the generalizability of our framework, we employ different LMMs as student models or teacher models and show the performance on seven downstream tasks. Specifically, Table 3 shows the performance of our framework when utilizing LLaVA-v1.5-7B as the student model, and LLaVA-72B, LLaMA-3.2-90B-Vision (Grattafiori et al., 2024)[11], GPT-4o-mini, or GPT-4o (Achiam et al., 2023) (`gpt-4o-2024-11-20`)[12] as the teacher model. Despite the capability gap between these teacher models on general multimodal tasks, their performance is quite comparable when utilizing them as the teacher model in our framework, demonstrating the generalizability and robustness of our framework.[13] Additionally, Table 4 shows the performance of our framework when using Qwen2-VL-7B as the student model and GPT-4o-mini as the teacher model. Note that Qwen2-VL-7B was the state-of-the-art LMM under 10B parameters at the time of submission. As we can see, though the pre-trained Qwen2-VL-7B has already significantly outperformed LLaVA-v1.5-7B across all downstream tasks, by employing our error-driven data-efficient tuning framework, its performance can be further improved by up to 4.30%, which further underscores the potential of our framework for effectively adapting state-of-the-art generic LLMs to specific downstream tasks.

4.3 Ablation Study

As shown in Table 5, we conduct ablation studies to demonstrate the effectiveness of each key component in our framework, using LLaVA-v1.5-7B as the student model, GPT-4o-mini as the teacher model, and Furniture Classification and Image Caption Match as the downstream tasks. We can see that: (1) Without the **Mistake Identification** module, performance drops by up to 4.70%, highlighting the challenge of directly analyzing missing skills from lengthy rationales; (2) Extracting skills from the entire validation dataset rather than error samples (i.e., **w/o Error Collection**), leads to a 7.6% performance drop at the 10K training scale, indicating inefficiency with limited training resources; (3) Using mistake steps as queries for targeted training samples retrieval (i.e., **w/o Skill Analysis**) results in a 7.90% performance drop, which is expected since the query used for data retrieval (i.e., mistake step) is not precisely aligned with the index of the supporting dataset (i.e., skills), though there is a correlation between them; (4)

[11]https://huggingface.co/meta-llama/Llama-3.2-90B-Vision-Instruct

[12]GPT-4o is approximately 16.7 times more expensive than GPT-4o-mini, which is why we opted not to use it in our main experiments.

[13]We further discuss the effectiveness of these teacher models in skill analysis in Appendix G.

Method	Teacher	# of Tuning Samples	MM-Bench	Appliance Cls	Furniture Cls	Living Thing Cls	VQA	Image-Cap Match	ScienceQA
Pre-trained LMM	-	0	64.30	45.80	49.00	79.40	77.00	64.10	65.34
Our Approach	LLaVA-72B	10K	63.55	62.00	64.40	89.00	75.50	75.00	64.90
Our Approach	LLaMA-90B	10K	63.62	61.90	64.50	89.20	75.50	76.40	65.89
Our Approach	GPT-4o-mini	10K	63.86	62.10	64.80	90.60	76.00	77.70	65.89
Our Approach	GPT-4o	10K	63.95	62.50	65.10	91.60	76.30	77.90	65.94
Our Approach	LLaVA-72B	100K	64.31	63.40	67.00	93.20	77.60	78.60	66.58
Our Approach	LLaMA-90B	100K	64.39	64.00	67.50	93.30	77.80	78.70	66.63
Our Approach	GPT-4o-mini	100K	64.41	64.10	67.70	93.60	79.00	80.10	68.02
Our Approach	GPT-4o	100K	64.50	64.80	68.00	93.60	79.10	81.10	68.07

Table 3: Evaluation results when using LLaVA-v1.5-7B as the student model and LLaVA-OneVision-72B, LLaMA-90B, GPT-4o-mini, and GPT-4o as different teacher models.

Method	# of Tuning Samples	MM-Bench	Appliance Cls	Furniture Cls	Living Thing Cls	VQA	Image-Cap Match	ScienceQA
Pre-trained LMM	0	82.80	63.70	67.60	93.60	87.90	84.30	85.50
Our Approach	10K	82.36	64.60	69.90	94.00	88.30	88.00	85.47
Our Approach	100K	**82.83**	**66.20**	**71.40**	**95.80**	**88.50**	**88.60**	**87.34**

Table 4: Evaluation results when using Qwen2-VL-7B as the student model and GPT-4o-mini as the teacher model.

Method	# of Tuning Samples	Furniture CIs	Image-Cap Match
Pre-trained LMM	0	49.00	64.10
Ours	10K	**64.80**	**77.70**
Ours w/o Mistake Identification	10K	63.90	73.00
Ours w/o Error Collection	10K	63.10	70.10
Ours w/o Skill Analysis	10K	62.30	69.80
Ours w/o Targeted Tuning	10K	60.60	65.03
Ours	30K	**67.10**	**80.00**
Ours w/o Mistake Identification	30K	65.90	78.30
Ours w/o Error Collection	30K	65.60	77.00
Ours w/o Skill Analysis	30K	64.70	74.30
Ours w/o Targeted Tuning	30K	63.13	71.97

Table 5: Ablation study with LLaVA-v1.5-7B as the student model and GPT-4o-mini as teacher model. (%)

Randomly sampling from the supporting dataset (**w/o Targeted Tuning**) also leads to consistent performance drops, confirming the importance of error-driven data selection for effective tuning.

4.4 Cost-Benefit Analysis

We perform a cost-benefit analysis using the ScienceQA and Image-Caption Matching tasks. Table 6 demonstrates that our framework incurs significantly lower overhead compared to the **Full Data** setting, leading to substantial reductions in tuning costs. To further optimize the cost-performance balance, the size of the validation set can be reduced (e.g., to 0.1K samples), accelerating the tuning process. Additionally, adopting open-source LLMs instead of proprietary API-based models can eliminate associated monetary costs while still maintaining robust performance gains over baseline methods.

4.5 Effectiveness of Mistake Identification

We further evaluate the effectiveness of our **Mistake Identification** method and compare it with three baselines: (1) **Random**, where an intermediate step is randomly selected as the mistake step; (2) **Prompt Per Step** (Tyen et al., 2024), where GPT-4o-mini is prompted to verify the correctness of each intermediate reasoning step, selecting the first incorrect one as the mistake step; (3) **Pseudo Rationale Match**, where GPT-4o-mini is first prompted to generate a sequence of pseudo reasoning steps based on the question and gold answer and compare them with the reasoning steps generated by the student model to find the mistake step. Due to the lack of gold labels for mistake steps in the validation datasets, we sample 100 error samples from the validation set of ScienceQA and manually label the mistake step for each error sample. The annotation process is detailed in Appendix C.1.

As shown in Table 7, the **Random** baseline achieves only 7.0% accuracy, reflecting the difficulty of mistake identification given that there are 15.22 reasoning steps per sample on average in the validation set. **Prompt Per Step** outperforms **Random**, though it tends to incorrectly mark steps as wrong when they cannot be directly inferred from previous ones. For example, given the following reasoning steps: "*Magnet sizes affect the magnitude of the magnetic force. Imagine magnets that are the same shape and material. The larger the magnets, the greater the magnetic force*.", **Prompt Per Step** identifies the second step as incorrect because "*The context doesn't indicate that they are all identical in shape or size. So this rationale step is incorrect.*". Instead, our mistake identification method surpasses all baselines by effectively analyzing the dynamics of the probabilities for each candidate answer from the teacher model, demonstrating its robustness.

Error Propagation We conduct additional experiments on ScienceQA to assess the impact of error propagation from the Mistake Identification stage on the model's overall performance. For this, we manually annotate mistake steps and missing skills in 100 error samples. These annotated missing

Method	Teacher	Task	$\mathcal{D}_{val}$	Mistake+Skill (s)	Data Retrieval (s)	Fine-tuning (s)	$\mathcal{D}_{train}$	Accuracy (%)	Complexity	API cost ($)
Pre-trained	-	ScienceQA	1K	-	-	-	-	65.34	-	0
Full Data	-	ScienceQA	1K	-	-	72,410	1,552K	67.23	$\mathcal{O}(\lvert\mathcal{D}_{support}\rvert)$	0
Ours	LLaVA-72B	ScienceQA	1K	224	50	4,353	100K	66.58	$\mathcal{O}(\lvert\mathcal{D}_{train}\rvert + \lvert\mathcal{D}_{error}\rvert * t)$	0
Ours	GPT-4o-mini	ScienceQA	0.1K	29	7	4,647	100K	66.83	$\mathcal{O}(\lvert\mathcal{D}_{train}\rvert + \lvert\mathcal{D}_{error}\rvert * t)$	0.1
Ours	GPT-4o-mini	ScienceQA	1K	219	54	4,041	100K	68.02	$\mathcal{O}(\lvert\mathcal{D}_{train}\rvert + \lvert\mathcal{D}_{error}\rvert * t)$	1.1
Pre-trained	-	Image-Cap Match	1K	-	-	-	-	64.10	-	0
Full Data	-	Image-Cap Match	1K	-	-	72,410	1,552K	84.70	$\mathcal{O}(\lvert\mathcal{D}_{support}\rvert)$	0
Ours	LLaVA-72B	Image-Cap Match	1K	177	48	4,534	100K	78.60	$\mathcal{O}(\lvert\mathcal{D}_{train}\rvert + \lvert\mathcal{D}_{error}\rvert * t)$	0
Ours	GPT-4o-mini	Image-Cap Match	0.1K	27	6	4,296	100K	78.50	$\mathcal{O}(\lvert\mathcal{D}_{train}\rvert + \lvert\mathcal{D}_{error}\rvert * t)$	0.1
Ours	GPT-4o-mini	Image-Cap Match	1K	198	50	4,391	100K	80.10	$\mathcal{O}(\lvert\mathcal{D}_{train}\rvert + \lvert\mathcal{D}_{error}\rvert * t)$	0.8

Table 6: Cost-benefit analysis on ScienceQA and Image-caption Matching tasks. $\|D_{support}\|$, $\|D_{train}\|$, $\|D_{val}\|$, $\|D_{error}\|$, and t represent the sample size of supporting dataset, training dataset, validation dataset, error set, and average reasoning steps, respectively. **Mistake+Skill**, **Data Retrieval**, and **Fine-tuning** indicate the runtime (in seconds) for Mistake Identification and Skill Analysis, Training Data Retrieval, and Fine-tuning stages, respectively, measured on $8 \times$ 40GB A100 GPUs. **API cost** represents the cost (in USD) for closed-source LMM.

Method	Accuracy (%)	Recall@3
Random	7.0	16.0
Prompt Per Step (Tyen et al., 2024)	28.0	34.0
Pseudo Rationale Match	59.0	68.0
Our Method	**65.0**	**77.0**

Table 7: Evaluation of various mistake identification methods on ScienceQA. Recall@3 quantifies the percentage of evaluation samples where the annotated gold mistake step matches the predicted step or falls within the three preceding steps.

skills are used to retrieve training samples for fine-tuning the student LMM, establishing the **Gold Mistake Step** setting. In the **Predicted Mistake Step** setting, we leverage the predicted mistake step from our method with about 65% accuracy for mistake identification. As shown in Table 8, our method can achieve comparable performance with **Gold Mistake Step** setting, indicating limited error propagation. This effectiveness is likely due to the fact that in 77% of the samples, as shown in Table 7, the annotated gold mistake step matches the predicted step or falls within the three preceding steps, allowing our approach to offer hints for identifying the mistake context across all reasoning steps even if the exact mistake step is not identified, thereby enhancing the skill analysis stage. Based on predicted mistake step, we leverage our skill analysis module to generate missing skills and manually verify their effectiveness. We find that in 87% of the samples, the generated missing skills match with the annotated gold missing skills, as detailed in Appendix G.

Setting	# of Tuning Samples	ScienceQA
Gold Mistake Step	10K	65.74
Predicted Mistake Step	10K	65.54
Gold Mistake Step	30K	67.07
Predicted Mistake Step	30K	66.48

Table 8: Impact of error propagation from mistake identification on the model's over performance.

δ	$lambda$	# Tuning	MI Accuracy (%)	ScienceQA Accuracy (%)
0.8	12	10K	60	64.80
0.2	0	10K	62	65.20
0	12	10K	63	65.64
0.2	12	10K	65	65.89
0.8	12	100K	60	66.53
0.2	0	100K	62	66.83
0	12	100K	63	67.18
0.2	12	100K	65	68.02

Table 9: Hyperparameter sensitivity analysis of the Mistake Identification (MI) method. **MI Accuracy** denotes the accuracy of correctly identifying the erroneous step, while **ScienceQA Accuracy** reflects the downstream task performance after tuning with the selected samples.

Hyperparameter Sensitivity We evaluate the robustness of the Mistake Identification (MI) method with respect to two key hyperparameters: the minimum probability gap threshold (δ) and the persistence window (λ), introduced in Section 3.3. The optimal values, $\delta = 0.2$ and $\lambda = 12$, are selected via a comprehensive grid search over $\delta \in \{0, 0.1, \ldots, 0.9\}$ and $\lambda \in \{0, 1, \ldots, 19\}$, using accuracy on the annotated validation set (see Appendix C.1) as the selection criterion. As shown in Table 9, variations around the optimal values lead to limited declines in both MI accuracy and downstream ScienceQA performance, suggesting that our method maintains practical robustness to changes in hyperparameters.

5 Conclusion

We propose a novel error-driven, data-efficient tuning paradigm to effectively adapt generic, pre-trained large multimodal models (LMMs) to various new and emerging downstream tasks without requiring any task-specific training samples. Extensive experiments show that our framework can significantly improve pre-trained LMM's performance on seven downstream tasks by retrieving targeted tuning samples from the supporting dataset. Future work can explore loss-driven latent skills (Xu et al., 2023c) to support more fine-grained skills.

Limitations

Though the extensive experiments have demonstrated the effectiveness of our error-driven data-efficient tuning framework, it still has several limitations: (1) **Requirement of Validation Set**. The task-specific validation set is crucial in our framework to measure the downstream task distribution and LMM's capability gaps. For certain tasks, even creating and labeling 1,000 samples could be expensive and time-consuming. Further research is necessary to remove the requirement of such task-specific validation sets. (2) **Mistake Identification Needs Further Improvement**. In this work, we develop a straightforward yet effective method for identifying mistakes within the rationales of LMMs. However, there is still potential for further enhancing this component, which is crucial for precisely analyzing the capability gaps of LMMs for target downstream tasks.

Ethics Statement

We carefully follow the ACM Code of Ethics [14] and have not found potential societal impacts or risks so far. To the best of our knowledge, this work has no notable harmful effects and uses, environmental impact, fairness considerations, privacy considerations, security considerations, or other potential risks.

Acknowledgements

This research is partially supported by the award No. 2238940 from the Faculty Early Career Development Program (CAREER) of the National Science Foundation (NSF), the U.S. DARPA ECOLE Program #HR001122S0052, and FoundSci Program #HR00112490370. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Afra Feyza Akyürek, Ekin Akyürek, Aman Madaan, Ashwin Kalyan, Peter Clark, Derry Wijaya, and Niket Tandon. 2023. RL4F: Generating Natural Language Feedback with Reinforcement Learning for Repairing Model Outputs. *arXiv*.

Shengnan An, Zexiong Ma, Zeqi Lin, Nanning Zheng, Jian-Guang Lou, and Weizhu Chen. 2023. Learning From Mistakes Makes LLM Better Reasoner. *arXiv*.

Athanasios Angelakis and Andrey Rass. 2024. A data-centric approach to class-specific bias in image data augmentation. *arXiv preprint arXiv:2403.04120*.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*.

Paul Bambrick-Santoyo. 2010. *Driven by data: A practical guide to improve instruction*. John Wiley & Sons.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, page 41–48, New York, NY, USA. Association for Computing Machinery.

Tom B Brown. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Gail A Carpenter and Stephen Grossberg. 1987. A massively parallel architecture for a self-organizing neural pattern recognition machine. *Computer vision, graphics, and image processing*, 37(1):54–115.

Hailin Chen, Amrita Saha, Steven Hoi, and Shafiq Joty. 2023a. Personalised Distillation: Empowering Open-Sourced LLMs with Adaptive Learning for Code Generation. *arXiv*.

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

Mayee F Chen, Nicholas Roberts, Kush Bhatia, Jue Wang, Ce Zhang, Frederic Sala, and Christopher Ré. 2023c. Skill-it! A Data-Driven Skills Framework for Understanding and Training Language Models. *arXiv*.

Ruibo Chen, Yihan Wu, Lichang Chen, Guodong Liu, Qi He, Tianyi Xiong, Chenxi Liu, Junfeng Guo, and Heng Huang. 2024. Your vision-language model itself is a strong filter: Towards high-quality instruction tuning with data selection. *arXiv preprint arXiv:2402.12501*.

[14]https://www.aclweb.org/portal/content/acl-code-ethics

Haixing Dai, Zhengliang Liu, Wenxiong Liao, Xiaoke Huang, Yihan Cao, Zihao Wu, Lin Zhao, Shaochen Xu, Wei Liu, Ninghao Liu, Sheng Li, Dajiang Zhu, Hongmin Cai, Lichao Sun, Quanzheng Li, Dinggang Shen, Tianming Liu, and Xiang Li. 2023. AugGPT: Leveraging ChatGPT for Text Data Augmentation. *arXiv*.

Sivan Doveh, Shaked Perek, M. Jehanzeb Mirza, Wei Lin, Amit Alfassy, Assaf Arbelle, Shimon Ullman, and Leonid Karlinsky. 2024. Towards multimodal in-context learning for vision & language models. *Preprint*, arXiv:2403.12736.

Ksenia Dmitrievna Dyatlova, Irina Mikchailovna Shvets, Elena Sergeevna Orlova, Yulia Vitalievna Sinitsyna, and Irina Valerievna Struchkova. 2018. Project-based learning as an instrument for the formation and development of research skills of biology students. In *Handbook of Research on Students' Research Competence in Modern Educational Contexts*, pages 132–150. IGI Global.

Yunzhen Feng, Elvis Dohmatob, Pu Yang, Francois Charton, and Julia Kempe. 2024. A tale of tails: Model collapse as a change of scaling laws. In *ICLR 2024 Workshop on Navigating and Addressing Data Problems for Foundation Models*.

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

Francis Green. 2011. *What is Skill?: An Inter-Disciplinary Synthesis*. Centre for Learning and Life Chances in Knowledge Economies and Societies London.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2021. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare (HEALTH)*, 3(1):1–23.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. *arXiv preprint arXiv:2004.10964*.

Guy Hacohen and Daphna Weinshall. 2019. On the power of curriculum learning in training deep networks. In *International conference on machine learning*, pages 2535–2544. PMLR.

Xiaochuang Han, Daniel Simig, Todor Mihaylov, Yulia Tsvetkov, Asli Celikyilmaz, and Tianlu Wang. 2023. Understanding in-context learning via supportive pretraining data. *arXiv preprint arXiv:2306.15091*.

Trevor Hastie, Andrea Montanari, Saharon Rosset, and Ryan J Tibshirani. 2022. Surprises in high-dimensional ridgeless least squares interpolation. *Annals of statistics*, 50(2):949.

Dorothée B Hoppe, Petra Hendriks, Michael Ramscar, and Jacolien van Rij. 2022. An exploration of error-driven learning in simple two-layer networks from a discriminative learning perspective. *Behavior Research Methods*, 54(5):2221–2251.

Yixing Jiang, Jeremy Irvin, Ji Hun Wang, Muhammad Ahmed Chaudhry, Jonathan H Chen, and Andrew Y Ng. 2024. Many-shot in-context learning in multimodal foundation models. *arXiv preprint arXiv:2405.09798*.

Krishnateja Killamsetty, Durga Sivasubramanian, Ganesh Ramakrishnan, Abir De, and Rishabh Iyer. 2021. GRAD-MATCH: Gradient Matching based Data Subset Selection for Efficient Deep Model Training. *arXiv*.

Tae-Hoon Kim and Jonghyun Choi. 2018. Screenernet: Learning self-paced curriculum for deep neural networks. *arXiv preprint arXiv:1801.00904*.

M Kumar, Benjamin Packer, and Daphne Koller. 2010. Self-paced learning for latent variable models. *Advances in neural information processing systems*, 23.

Po-Nien Kung, Fan Yin, Di Wu, Kai-Wei Chang, and Nanyun Peng. 2023. Active Instruction Tuning: Improving Cross-Task Generalization by Training on Prompt Sensitive Tasks. *arXiv*.

Changho Lee, Janghoon Han, Seonghyeon Ye, Stanley Jungkyu Choi, Honglak Lee, and Kyunghoon Bae. 2024a. Instruction matters, a simple yet effective task selection approach in instruction tuning for specific tasks. *arXiv preprint arXiv:2404.16418*.

Nicholas Lee, Thanakul Wattanawong, Sehoon Kim, Karttikeya Mangalam, Sheng Shen, Gopala Anumanchipali, Michael W Mahoney, Kurt Keutzer, and Amir Gholami. 2024b. Llm2llm: Boosting llms with novel iterative data enhancement. *arXiv preprint arXiv:2403.15042*.

Yong Jae Lee and Kristen Grauman. 2011. Learning the easy things first: Self-paced visual category discovery. In *CVPR 2011*, pages 1721–1728. IEEE.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. 2024a. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*.

Fei-Fei Li, Marco Andreeto, Marc'Aurelio Ranzato, and Pietro Perona. 2022. Caltech 101.

Haoran Li, Qingxiu Dong, Zhengyang Tang, Chaojun Wang, Xingxing Zhang, Haoyang Huang, Shaohan Huang, Xiaolong Huang, Zeqiang Huang, Dongdong Zhang, et al. 2024b. Synthetic data (almost) from scratch: Generalized instruction tuning for language models. *arXiv preprint arXiv:2402.13064*.

Ming Li, Yong Zhang, Shwai He, Zhitao Li, Hongyu Zhao, Jianzong Wang, Ning Cheng, and Tianyi Zhou. 2024c. Superfiltering: Weak-to-strong data filtering for fast instruction-tuning. *arXiv preprint arXiv:2402.00530*.

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

Wenyan Li, Jonas F Lotz, Chen Qiu, and Desmond Elliott. 2023b. Data Curation for Image Captioning with Text-to-Image Generative Models. *arXiv*.

Zhenwen Liang, Wenhao Yu, Tanmay Rajpurohit, Peter Clark, Xiangliang Zhang, and Ashwin Kalyan. 2023. Let GPT be a math tutor: Teaching math word problem solvers with customized exercise generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14384–14396, Singapore. Association for Computational Linguistics.

Chi-Heng Lin, Chiraag Kaushik, Eva L Dyer, and Vidya Muthukumar. 2024. The good, the bad and the ugly sides of data augmentation: An implicit spectral regularization perspective. *Journal of Machine Learning Research*, 25(91):1–85.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2024a. Visual instruction tuning. *Advances in neural information processing systems*, 36.

Liangxin Liu, Xuebo Liu, Derek F Wong, Dongfang Li, Ziyi Wang, Baotian Hu, and Min Zhang. 2024b. Selectit: Selective instruction tuning for large language models via uncertainty-aware self-reflection. *arXiv preprint arXiv:2402.16705*.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, Kai Chen, and Dahua Lin. 2024c. Mmbench: Is your multi-modal model an all-around player? *Preprint*, arXiv:2307.06281.

Zikang Liu, Kun Zhou, Wayne Xin Zhao, Dawei Gao, Yaliang Li, and Ji-Rong Wen. 2024d. Less is More: Data Value Estimation for Visual Instruction Tuning. *arXiv*.

Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. 2022. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *The 36th Conference on Neural Information Processing Systems (NeurIPS)*.

Renqian Luo, Liai Sun, Yingce Xia, Tao Qin, Sheng Zhang, Hoifung Poon, and Tie-Yan Liu. 2022. Biogpt: generative pre-trained transformer for biomedical text generation and mining. *Briefings in bioinformatics*, 23(6):bbac409.

Fan Ma, Deyu Meng, Qi Xie, Zina Li, and Xuanyi Dong. 2017. Self-paced co-training. In *International Conference on Machine Learning*, pages 2275–2284. PMLR.

Tambet Matiisen, Avital Oliver, Taco Cohen, and John Schulman. 2017. Teacher-Student Curriculum Learning. *arXiv*.

Vidya Muthukumar, Kailas Vodrahalli, Vignesh Subramanian, and Anant Sahai. 2020. Harmless interpolation of noisy data in regression. *IEEE Journal on Selected Areas in Information Theory*, 1(1):67–83.

Nihal V Nayak, Yiyang Nan, Avi Trost, and Stephen H Bach. 2024. Learning to generate instruction tuning datasets for zero-shot task adaptation. *arXiv preprint arXiv:2402.18334*.

Pascual Pérez-Paredes and María Sánchez-Tornel. 2009. Understanding e-skills in the flt context. In *Handbook of Research on E-Learning Methodologies for Language Acquisition*, pages 1–21. IGI Global.

Jeff M Phillips. 2016. Coresets and Sketches. *arXiv*.

Jingyuan Qi, Zhiyang Xu, Ying Shen, Minqian Liu, Di Jin, Qifan Wang, and Lifu Huang. 2023. The art of socratic questioning: Recursive thinking with large language models. *Preprint*, arXiv:2305.14999.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Revant Gangi Reddy, Xilin Rui, Manling Li, Xudong Lin, Haoyang Wen, Jaemin Cho, Lifu Huang, Mohit Bansal, Avirup Sil, Shih-Fu Chang, et al. 2022. Mumuqa: Multimedia multi-hop news question answering via cross-media knowledge extraction and grounding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11200–11208.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. 1986. Learning representations by back-propagating errors. *nature*, 323(6088):533–536.

Ozan Sener and Silvio Savarese. 2017. Active learning for convolutional neural networks: A core-set approach. *arXiv preprint arXiv:1708.00489*.

Ying Shen, Zhiyang Xu, Qifan Wang, Yu Cheng, Wenpeng Yin, and Lifu Huang. 2024. Multimodal instruction tuning with conditional mixture of lora. *arXiv preprint arXiv:2402.15896*.

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Yarin Gal, Nicolas Papernot, and Ross Anderson. 2023. The curse of recursion: Training on generated data makes models forget. *arXiv preprint arXiv:2305.17493*.

Valentin I Spitkovsky, Hiyan Alshawi, and Dan Jurafsky. 2010. From baby steps to leapfrog: How "less is more" in unsupervised dependency parsing. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 751–759.

Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J Gordon. 2018. An Empirical Study of Example Forgetting during Deep Neural Network Learning. *arXiv*.

Gladys Tyen, Hassan Mansoor, Victor Cărbune, Yuanzhu Peter Chen, and Tony Mak. 2024. Llms cannot find reasoning errors, but can correct them given the error location. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 13894–13908.

Danqing Wang and Lei Li. 2023. Learning from mistakes via cooperative study assistant for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10667–10685.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. 2024. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.

Sijia Wang and Lifu Huang. 2024. Targeted augmentation for low-resource event extraction. *arXiv preprint arXiv:2405.08729*.

Yizhong Wang, Hamish Ivison, Pradeep Dasigi, Jack Hessel, Tushar Khot, Khyathi Chandu, David Wadden, Kelsey MacMillan, Noah A Smith, Iz Beltagy, et al. 2023. How far can camels go? exploring the state of instruction tuning on open resources. *Advances in Neural Information Processing Systems*, 36:74764–74786.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022a. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models. *Preprint*, arXiv:2201.11903.

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

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. 2024a. Less: Selecting influential data for targeted instruction tuning. *arXiv preprint arXiv:2402.04333*.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. 2024b. LESS: Selecting Influential Data for Targeted Instruction Tuning. *arXiv*.

Sang Michael Xie, Shibani Santurkar, Tengyu Ma, and Percy S Liang. 2023. Data selection for language models via importance resampling. *Advances in Neural Information Processing Systems*, 36:34201–34227.

Wenda Xu, Daniel Deutsch, Mara Finkelstein, Juraj Juraska, Biao Zhang, Zhongtao Liu, William Yang Wang, Lei Li, and Markus Freitag. 2023a. Pinpoint, Not Criticize: Refining Large Language Models via Fine-Grained Actionable Feedback. *arXiv*.

Zhiyang Xu, Chao Feng, Rulin Shao, Trevor Ashby, Ying Shen, Di Jin, Yu Cheng, Qifan Wang, and Lifu Huang. 2024. Vision-flan: Scaling human-labeled tasks in visual instruction tuning. *arXiv preprint arXiv:2402.11690*.

Zhiyang Xu, Jay Yoon Lee, and Lifu Huang. 2023b. Learning from a friend: Improving event extraction via self-training with feedback from abstract meaning representation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 10421–10437.

Zhiyang Xu, Ying Shen, and Lifu Huang. 2022. Multiinstruct: Improving multi-modal zero-shot learning via instruction tuning. *arXiv preprint arXiv:2212.10773*.

Zifan Xu, Haozhu Wang, Dmitriy Bespalov, Peter Stone, and Yanjun Qi. 2023c. Latent Skill Discovery for Chain-of-Thought Reasoning. *arXiv*.

Zeyuan Yang, Peng Li, and Yang Liu. 2023. Failures Pave the Way: Enhancing Large Language Models through Tuning-free Rule Accumulation. *arXiv*.

Barry Menglong Yao, Yu Chen, Qifan Wang, Sijia Wang, Minqian Liu, Zhiyang Xu, Licheng Yu, and Lifu Huang. 2023. Ameli: Enhancing multimodal entity linking with fine-grained attributes. *arXiv preprint arXiv:2305.14725*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2024. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36.

Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. 2023. Mm-vet: Evaluating large multimodal models for integrated capabilities. *Preprint*, arXiv:2308.02490.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. 2024. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. *Preprint*, arXiv:2311.16502.

Min Zhang, Zhongwei Yu, Hai Wang, Hongbo Qin, Wei Zhao, and Yan Liu. 2019. Automatic digital modulation classification based on curriculum learning. *Applied Sciences*, 9(10):2171.

Chenyang Zhao, Xueying Jia, Vijay Viswanathan, Tongshuang Wu, and Graham Neubig. 2024a. Self-guide: Better task-specific instruction following via self-synthetic finetuning. *arXiv preprint arXiv:2407.12874*.

Haozhe Zhao, Zefan Cai, Shuzheng Si, Xiaojian Ma, Kaikai An, Liang Chen, Zixuan Liu, Sheng Wang, Wenjuan Han, and Baobao Chang. 2024b. Mmicl: Empowering vision-language model with multi-modal in-context learning. *Preprint*, arXiv:2309.07915.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. 2024. Lima: Less is more for alignment. *Advances in Neural Information Processing Systems*, 36.

Yuke Zhu, Oliver Groth, Michael Bernstein, and Li Fei-Fei. 2016. Visual7w: Grounded question answering in images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4995–5004.

A Answer Format

Table 10 shows the answer formats that we use to parse the answer.

B Prompt Template

B.1 Inference Prompt Template

Figure 3 shows the Inference Prompt Template.

B.2 Mistake Identification Prompt Template

Figure 4 shows Mistake Identification Prompt Template

B.3 Skill Analysis Prompt Template

Figure 5 shows Skill Analysis Prompt Template

B.4 Skill Set Analysis Prompt Template

Figure 6 shows Skill Set Analysis Prompt Template.

C Mistake Identification

C.1 Human Annotation for Mistake Identification

We first run the student model on the validation set of ScienceQA dataset and obtain error samples as we mentioned in Section 3.2. We then randomly select 100 error samples for annotation. For each error sample, we split the student model's rationale into a sequence of reasoning steps[15]. The annotator will then annotate these error samples following the following guidelines:

- Open one of your annotation web pages
- For each sample, check through the question, the choices, the image, the correct answer, and the wrong prediction.
- Then you will read rationale step one by one and check whether the current rationale step contains logical errors. If yes, you can record the corresponding index (starting from 0).
- If you did not record any rationale step after checking all of them, you can provide "-1" as the label of mistake step for this sample.

C.2 Probability Gap Statistic

Figure 7 shows probability gap statistics in our annotated validation set, where Probability Gap = p(student model's wrong answer) - p(correct answer).

[15]Following previous studies (Tyen et al., 2024), we treat each sentence in the rationale as one reasoning step.

D Definition and Explanation of skills

In the education domain, skill is defined as an ability to carry out a task with pre-determined results, often within a given amount of time, energy, or both (Dyatlova et al., 2018). Some studies stress out the expandability of skill: skill refers to any ability acquired by training or practice, allowing individuals to perform well in multifarious types of tasks (Pérez-Paredes and Sánchez-Tornel, 2009; Green, 2011). In this work, we follow (Chen et al., 2023c) and define a skill s as a unit of behavior with associated data X such that if the LMM is trained on dataset D, where $D \subseteq X$, it has improved performance on samples belonging to $X \backslash D$. This definition of a skill is flexible—it focuses on the expandability of skill and means that given a training dataset associated with the skill, a model f has an improved performance when evaluated on validation data associated with this skill. Under this definition, a skill could be a fine-grained, instance-specific ability like "Identify the poles of a magnet", instead of general skills like "color recognition", "shape recognition", and "texture recognition".

E Experiment Results for Random Sampling

For the Random Sampling baseline, the random sampling process was repeated three times and we report detailed results for each run in Table 11.

F Analysis of Task-specific Performance Variations

Table 12 reveals a strong negative correlation between the number of reasoning steps and the performance gains from our framework. We observe that tasks involving fewer reasoning steps typically achieve greater performance improvements, whereas tasks with longer reasoning chains exhibit comparatively limited gains. We attribute this pattern to two primary factors: (a) Error Localization Complexity—the challenge of accurately identifying the erroneous reasoning step intensifies as reasoning chains grow longer; and (b) Inherent Task Difficulty—tasks requiring longer reasoning chains are inherently more complex, thus making them more challenging targets for performance enhancement.

Answer Format	Regular Expression Pattern
Answer is (A)	(?i)answer is \(([A-Z])
Answer is (A	(?i)answer is \(([A-Z])
Answer is A.	(?i)answer is ([A-Z])\.
Answer: A	(?i)answer:\s?([a-z])
A is the correct answer	(?i)([A-Z]) is the correct
A	(?<!\S)[a-zA-Z](?!\S)(?!.*[a-zA-Z])
answer is the option A	(?<!\S)[a-zA-Z] (?!\S)(?!.*[a-zA-Z])
choose the answer, A	(?i)choose the answer,\s?([a-z])

Table 10: Answer format table

Inference Prompt Template

Question: Is a violin a good or a service?
Choices: (A) a good. (B) a service.
Rationale: To decide whether a violin is a good or a service, ask these questions: Is a violin something you can touch? Yes. Is a violin a job you might pay someone else to do? No. So, a violin is a good. **The final answer is A**.

Question:Will these magnets attract or repel each other?
Choices: (A) attract (B) repel
Let us think step by step. Provide your Rationale and the final answer. The final answer should be the option's letter from the given choices.

Figure 3: One example prompt for ScienceQA task to obtain the student model's prediction.

Mistake Identification Prompt Template

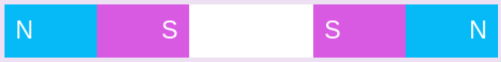
{***few-shot demonstrations***}
Question:Will these magnets attract or repel each other?
Choices: (A) attract (B) repel
Prior Knowledge: There is a probability of 60% that these magnets repel each other.
Rationale:
Identify the magnets: There are two magnets in the image.
Determine their polarity: we see two magnets placed end to end, with **one magnet's "S" end facing the other magnet's "N" end**.
Answer with the option's letter from the given choices directly. Please provide the answer without explanation. If you can not find the correct answer, then guess based on the Prior Knowledge. Please provide the answer in the format of 'The answer is A/B/C/D/E'

Figure 4: One example prompt to obtain the teacher model's prediction by following the student model's rationale steps. We then identify the mistake rationale step based on the evolution in probabilities of predicted options from the teacher model.

Method	# of Tuning Samples	MM-Bench	Appliance Cls	Furniture Cls	Living Thing Cls	VQA	Image-Cap Match	ScienceQA
Random 1	10K	63.40	57.70	61.00	85.60	74.80	63.20	64.06
Random 2	10K	62.80	55.30	60.30	79.10	73.00	66.30	63.11
Random 3	10K	62.35	59.40	60.50	81.60	74.30	65.60	63.81
Random 1	30K	62.65	61.10	63.60	87.90	77.10	73.50	63.01
Random 2	30K	62.20	61.30	62.30	86.60	75.40	74.10	63.96
Random 3	30K	62.95	60.80	63.50	86.00	74.00	68.30	63.16
Random 1	100K	62.95	61.20	66.30	91.00	77.10	78.30	65.74
Random 2	100K	63.86	62.00	66.70	87.60	76.30	76.40	64.55
Random 3	100K	62.05	59.30	65.70	87.40	77.30	77.80	63.36

Table 11: Evaluation results on seven downstream tasks with different numbers of tuning samples retrieved from the supporting dataset. (%).

Skill Analysis Prompt Template

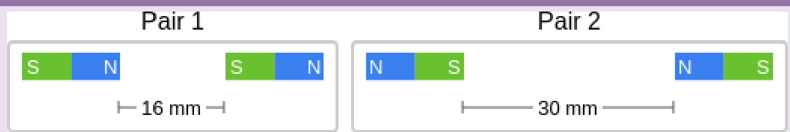


{***few-shot demonstrations***}
Question:Will these magnets attract or repel each other?
Choices: (A) attract (B) repel
Correct Answer: (B) repel
Rationale:
Identify the magnets: There are two magnets in the image.
Determine their polarity: we see two magnets placed end to end, with one magnet's "S" end facing the other magnet's "N" end.
Understand the interaction between magnets: Opposite poles attract each other, while like poles repel
Apply the knowledge to the question: Since the north pole of one magnet is facing the south pole of the other magnet, they will attract each other.
Wrong Rationale Step:
Determine their polarity: we see two magnets placed end to end, with one magnet's "S" end facing the other magnet's "N" end.

Please refer to Demonstration to find the Missing Skill. Each Question comes with its Options, and Correct Answer. The Rationale Steps and the specific Wrong Rationale Step will be provided to you. Please analyze the Missing Skill based on the Wrong Rationale Steps.

Figure 5: One example prompt to trigger the teacher model to analyze the missing skill based on the wrong rationale step.

Skill Set Analysis Prompt Template



{***few-shot demonstrations***}
Question: Think about the magnetic force between the magnets in each pair. Which pair has the stronger magnetic force?
Context: The images show two pairs of magnets. The magnets in different pairs do not affect each other. All the magnets shown are made of the same material.
Answer: The magnetic force is stronger in Pair 1

Based on the question, answer, and the image, could you provide a detailed analysis of the skills required to answer the questions? Provide some skills that are specific to the questions instead of general skills like reasoning or observation skills.

Figure 6: One example prompt to trigger the teacher model to analyse a sequence of required skills for each sample in the supporting dataset.

G Effectiveness of Different Teacher Models

We further discuss whether the choice of the teacher model affects the effectiveness of the skill analysis. We first ask the annotator to write the gold missing skills based on question, reasoning steps, and answer for 100 error samples. We then leverage two teacher models, GPT-4o-mini and LLaVA 72B, to predict missing skills and ask the annotator to manually compare these predicted missing skills with gold missing skills. Our experiments indicate that the missing skills generated by GPT-4o-mini align with the annotated gold missing skills in 87% of the samples, whereas those generated by LLaVA-72B match the annotated gold missing skills in 79% of the samples.

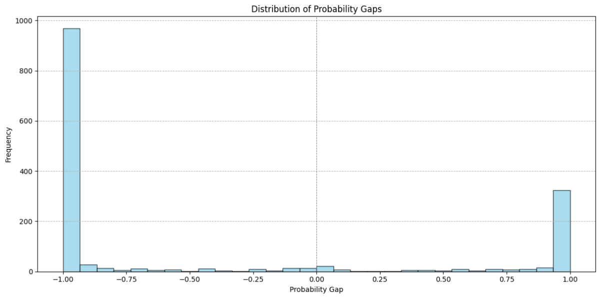


Figure 7: Probability gap statistics in our annotated validation set, where Probability Gap=p(student model's wrong answer)-p(correct answer).

Task	# of Reasoning Step	Performance Gain
Appliance CIs	7.64	18.30
MM-Bench	13.23	0.11
Furniture CIs	8.33	18.70
Living Thing CIs	6.27	14.20
VQA	12.74	2.00
Image-Cap Match	7.13	16.00
ScienceQA	15.00	2.68

Table 12: Task characteristics for all evaluation tasks. **# of Reasoning Steps** indicates the average number of reasoning steps, and **Performance Gain** refers to the improvement compared with the pre-trained LLaVA-v1.5-7B.

H Experiment Details

We conduct experiments on 8×40GB A100 GPUs. In the 100K training sample setting, one training can run for 2 hours. We use learning rate as 2×10^{-4} and batch size as 128.