Teaching-Assistant-in-the-Loop: Improving Knowledge Distillation from Imperfect Teacher Models in Low-Budget Scenarios

Yuhang Zhou
University of Maryland
College Park, USA
tonyzhou@umd.edu

Wei Ai
University of Maryland
College Park, USA
aiwei@umd.edu

Abstract

There is increasing interest in distilling task-specific knowledge from large language models (LLM) to smaller student models. Nonetheless, LLM distillation presents a dual challenge: 1) there is a high cost associated with querying the teacher LLM, such as GPT-4, for gathering an ample number of demonstrations; 2) the teacher LLM might provide imperfect outputs with a negative impact on the student's learning process. To enhance sample efficiency within resource-constrained, imperfect teacher scenarios, we propose a three-component framework leveraging three signal types. The first signal is the student's self-consistency (consistency of student multiple outputs), which is a proxy of the student's confidence. Specifically, we introduce a "teaching assistant" (TA) model to assess the uncertainty of both the student's and the teacher's outputs via confidence scoring, which serves as another two signals for student training. Furthermore, we propose a two-stage training schema to first warm up the student with a small proportion of data to better utilize student's signal. Experiments have shown the superiority of our proposed framework for four complex reasoning tasks. On average, our proposed two-stage framework brings a relative improvement of up to 20.79% compared to fine-tuning without any signals across datasets.

1 Introduction

Large language models (LLMs) have demonstrated state-of-the-art (SOTA) performance across a wide spectrum of tasks, and their efficacy is primarily attributed to their substantial model size, enabling them to possess capabilities that smaller models lack (Brown et al., 2020; Raffel et al., 2020; Chowdhery et al., 2022; Touvron et al., 2023). In most cases, a pre-trained LLM undergoes several specialized fine-tuning stages, such as instruction tuning, self-supervised tuning, and reinforcement learning with human feedback (Touvron et al., 2023; OpenAI, 2023), in order to excel at downstream tasks. Nevertheless, fine-tuning LLMs are challenging, primarily due to the demanding computational resources required (Wei et al., 2022; Chowdhery et al., 2022) and the limited access to LLMs, such as GPT-4 and Claude (Ouyang et al., 2022; OpenAI, 2023) prevents further fine-tuning.

To resolve the above difficulties, knowledge distillation (KD) uses the outputs of a larger LLM (Teacher) to train a smaller model (Student), such as GPT-J-6B or LLaMa-7B (Wang and Komatsuzaki, 2021; Touvron et al., 2023). KD has gained significant attention and led to models such as Alpaca (Taori et al., 2023) and Vicuna (Chiang et al., 2023). The KD pipeline first utilizes In-Context Learning (ICL) on the teacher model to generate outputs, forming distillation sets and then use the teacher generations to fine-tune the student model.

However, current work of distilling LLMs presents two major difficulties: First, it can be prohibitively expensive to collect sufficiently large distillation sets, especially when querying proprietary LLMs like GPT-4, so a budget of asking the teacher is required in most real-life scenarios; Second, as LLMs may not have seen task-specific data, the quality of their demonstrations might be low. For instance, the zero-shot accuracy of InstructGPT on the Aqua task is only 34.25% (Ouyang et al., 2022), and these suboptimal demonstrations negatively affect the student's performance, as elaborated in Section 4. Effectively excluding flawed annotations, especially for unlabeled sets, is essential.

To address these KD challenges, we introduce a novel three-component KD framework for LLMs to learn efficiently from an imperfect teacher within budget constraints. In addition to the student and the teacher (as in standard KD pipeline), our framework introduces a Teaching Assistant (TA) model which mimics an TA in the real-life class and acts as an intermediary to communicate with students and teachers. The TA can have a larger size and different architecture than the student, but smaller size

Findings of the Association for Computational Linguistics: ACL 2024, pages 265–282
August 11-16, 2024 ©2024 Association for Computational Linguistics

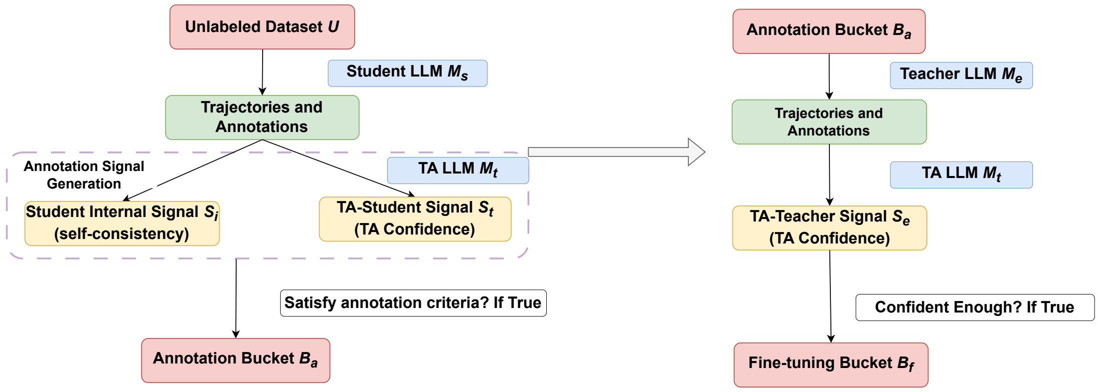

Figure 1: **Overview of the proposed three-component KD framework.** The student model M_s first makes the inference on the unlabeled dataset U and we calculate student internal signal s_i and TA-student signal s_t. Next, we distinguish whether this sample is worth annotating and add the satisfied ones to the annotation bucket B_a. Then, the teacher model M_e will annotate the examples in B_a. If TA-teacher signal s_e of the TA model is confident in the teacher's answer, we will add this question and annotations to the fine-tuning bucket B_f.

than the teacher, leading to significantly reduced inference cost compared to the teacher.

With the help of the TA model, our framework utilizes three types of new signals to refine the distillation. First, the student internal signal (self-consistency score) gauges the student's confidence and determines whether it should be "forwarded" to the teacher for annotation. Second, the TA model independently generates its TA-student signal for a question, aiding in the decision to annotate. Lastly, the TA model assesses annotations from the teacher (TA-teacher signal), deciding whether they merit inclusion in the student model's training dataset.

Moreover, the signal from the student model on a complex task is of lower quality for active selection. Therefore, we propose an extension of the two-stage training. Initially, we allocate 10% of the annotation budget to fine-tune the student. After this "warm-up" stage, we utilize the remaining 90% of the annotation budget. The student in both stages are fine-tuned within the proposed framework. Our experiments demonstrate that this approach further improves performance in various tasks.
Our work introduces three main contributions:

- We introduce a TA model and develop a three-component KD framework that leverages three signals to improve the sample efficiency given a annotation budget and imperfect teacher LLM.
- We further improve the three-component KD framework by introducing a two-stage training approach.
- We conduct extensive experiments on various datasets and public models and the proposed two-stage framework increases performance by up to 20.79%, compared to fine-tuning without signals, suggesting the effectiveness of our framework.

2 Related Work

2.1 Language model prompting

Prompt-based learning refers to using prompts to induce the embedded knowledge in the language models to complete downstream tasks (Radford et al., 2019; Liu et al., 2023b; Raffel et al., 2020). Among various prompting methods, in-context learning (ICL) through adding a few demonstrations and instructions in the prompt to elicit the correct answer has been shown as an effective approach (Brown et al., 2020). Since the output from language models is sensitive to the instruction composition and demonstration selection, how to tune and design the prompts for ICL has attracted the attention of many researchers (Liu et al., 2023b). Chain-of-thought (CoT) prompting feeds the step-by-step examples and specifically designed instructions to ask the language models to generate the intermediate reasoning step of a complex task (Wei et al., 2022). By grounding on the reasoning steps, the language models perform better on a wide variety of complex tasks. Based on CoT methods, researchers also divide the reasoning process into multiple steps (Creswell and Shanahan, 2022; Creswell et al., 2022). Moreover, Yao et al. (2022) propose another prompting method to integrate the additional action step to obtain external knowledge, which has been shown to be more effective in reasoning and decision making tasks.

2.2 Signals and feedback in language models

Besides asking the LLM to generate more intermediate steps, some works utilize the signals or feedback generated from the language model output to continually refine the prompts and resolve the hallucination (Madaan et al., 2023; Shinn et al., 2023; Diao et al., 2023; Su et al., 2022; Peng et al., 2023; Welleck et al., 2022; Liu et al., 2023a; Li et al., 2024b). Diao et al. (2023) and Su et al. (2022) use the measured uncertainty or score to actively select the uncertain exemplars on the prompts to increase ICL performance. Madaan et al. (2023); Shinn et al. (2023) rely on the language model itself to compose the natural language feedback. These signal-based methods are usually utilized in language model prompting (ICL) without any model parameter updates, which are distinct from our work since we designed signals for collecting high-quality examples for model finetuning.

2.3 Knowledge distillation in language models

Knowledge Distillation (KD) can be framed as training small student models based on data generated from large teacher models to reduce model size and expense while retaining capabilities (Sanh et al., 2019; Hinton et al., 2015; Buciluǎ et al., 2006). For the KD pipeline with black-box LLMs, there are two lines of work. The first is to ask teacher models to generate the final answers and to do standard fine-tuning on the final answers (Yoo et al., 2021; Schick and Schütze, 2020, 2021; Zhou et al., 2023). Another line of work is to ask the student model to fine-tune the teacher-generated CoT or other prompting trajectories (rationales) (Yao et al., 2022; Ho et al., 2022; He et al., 2023; Shridhar et al., 2023; Hsieh et al., 2023; Wang et al., 2023; Feng et al., 2023). Through the developed prompting methods, researchers find that sequence-level distillation of reasoning steps generated by the teacher is more effective (Yao et al., 2022; Ho et al., 2022), enabling the reasoning or decision-making abilities in students. However, most sequence-level KD works utilize the known correct or incorrect trajectories to fine-tune the student (Li et al., 2023; Wang et al., 2023; Chen et al., 2023; Liu et al., 2023c; An et al., 2023). Moreover, they consider the scenario with sufficiently large fine-tuning set while the fixed teacher budget and the flawed teacher annotations in the unlabeled data are yet to be explored. Yu et al. (2023); Liang et al. (2023) boost the fine-tuning set by rephrasing noisy rationales but do not actively select annotated samples to save teacher budget. Mirzadeh et al. (2020); Son et al. (2021) also leverage Teacher Assistant (TA) models for KD tasks but within the realm of computer vision, diverging from our approach that applies TA models in sequence-level KD tasks. Table 1 summarizes the similarity and uniqueness of our work with previous related ICL or KD approaches.

Method	Model Updates	Selective Samples	Teacher Usage	w/o GT Labels
Zero-shot (Radford et al., 2019)	✗	✗	✗	✓
Few-shot CoT (Wei et al., 2022)	✗	✗	✗	✗
Active prompting (Diao et al., 2023)	✗	✓	✓	✓
Finetune CoT (Ho et al., 2022)	✓	✗	✓	✗
Distilling step-by-step (Hsieh et al., 2023)	✓	✗	✓	✗
MetaMath (Yu et al., 2023)	✓	✗	✓	✗
Finetune ReAct (Yao et al., 2022)	✓	✗	✓	✗
TA-in-the-loop (our work)	✓	✓	✓	✓

Table 1: A comparison of our work to closely related prior ICL or KD approaches for LLMs.

3 Methodology

Figure 1 shows the overview of our framework. Our three-component framework consists of two pipelines: 1): Collect a fixed budget of examples that meet annotation criteria and add them to the bucket B_a: the intuition here is that we want to choose the challenging examples which the student model has a chance to learn correctly instead of choosing examples randomly that will include oversimple or complicated examples beyond the student's learning ability 2): Annotate the examples from B_a and return a high-quality fine-tuning set B_f: we want to filter out low-quality or wrong annotations generated by the teacher LLM.

We leverage the interactions among the student model M_s, the teacher model M_e, and the TA model M_t to generate three types of signals and achieve the above goals. For a given example, we will use the student internal signal s_i as well as the TA-student signal from the TA model s_t to decide whether the example should be added to B_a. After collecting the fixed budget of samples for B_a, we ask the teacher model to annotate those examples and apply the TA-teacher signal s_e from the TA model to filter out low-quality demonstrations. We introduce these three signals in detail below, and the algorithms outlining this procedure are also presented in Algorithm 1.

3.1 Signals for annotation

We propose two types of signals to decide whether an example meets the annotation criteria: student internal signal s_i and TA-student signal s_t.

Algorithm 1 **Three-component Knowledge Distillation Framework**

```
Input: Unlabeled dataset U, student model M_s, TA model M_t, teacher model M_e, confidence prompt P_c, annotation criteria annotate(·), confidence set C, annotation bucket B_a, fine-tuning bucket B_f
Output: The desired fine-tuning bucket B_f
for each input x from dataset U do
    t_i(x) = M_s(x, P_i) for i ∈ {1, ··· , n}
    s_i(x) = Uniq(t_1(x) ... t_n(x))
    s_t(x) = M_t(x, t_1, P_c)
    if annotate(s_i, s_t) is True then
        Add x into annotate bucket B_a
for each input x from dataset B_a do
    t_e(x) = M_e(x, P_i)
    s_e(x) = M_t(x, t_e, P_c)
    if s_e in C then
        Add x into fine-tuning bucket B_f
return B_f
```

3.1.1 Student internal signal s_i

Motivated by self-consistency work (Wang et al., 2022), we utilize the disagreement number (self-consistency number) as the measurement of the uncertainty of student models. For a given sample x and prompt P_i, we use stochastic temperature sampling with a fixed temperature and repeat the process for n times with answers $t_1, \cdots t_n$:

$$t_i(x) = M_s(x, P_i) \text{ for } i \in \{1, \cdots, n\}$$

then the internal signal can be calculated by counting the unique values among n answers

$$s_i(x) = \text{Uniq}(t_1(x), \cdots, t_n(x))$$

where Uniq is counting the unique answers from t_1 to t_n, and $s_i \in \{1, 2, \cdots, n\}$. In Table 14 in the Appendix, we present two examples with different numbers of student disagreement.

3.1.2 TA-Student signal s_t

We also utilize the TA model as an auxiliary signal to characterize the uncertainty of student's generations, given the observation that the confidence estimated by a LLM itself is prone to be overconfident (Diao et al., 2023; Si et al., 2022). For a given example x and the student annotation t (rationales and answer), the TA model is provided with a crafted confidence prompt P_c to classify the student annotation t

$$s_t(x) = M_t(x, t, P_c)$$

where s_t is a categorical variable belonging to the confidence set {very confident, confident, not confident, wrong answer}. Details and examples of the confidence prompt P_c and t are shown in Table 12 and 13 in Appendix.

3.1.3 Annotation criteria

Our annotation criteria are formulated by the student internal signal s_i and TA-student signal s_t. We compute the complexity score of an example x:

$$c(x) = \alpha \mathbb{1}_{s_i(x) \in C_1} + \beta \mathbb{1}_{s_t(x) \in C_2} \quad (1)$$

where $\alpha, \beta \in \{0, 1\}$ are the signal weights. When $\alpha = 1$ and $\beta = 1$, it means both signal s_i and s_t are utilized, and conversely if either α or β is set to 0, the corresponding signal is not employed. $\mathbb{1}$ is the indicator function and C_1, C_2 are complexity sets to actively select the examples that bring the deepest learning curve for the student model. The example will be added to the annotation bucket B_a if $c(x) \geq \alpha + \beta$ until the predefined budget is met.

In our experiment setup, we use $n = 5$, $C_1 = \{2, 3\}$ and $C_2 = \{\text{confident, not confident}\}$ after the preliminary exploration about hyperparameter combinations (Appendix B.4), which ensures that we can choose questions that are not too easy or too hard to learn for the student model.

3.2 TA-Teacher signal s_e

With the desired annotation bucket, we utilize the teacher model M_e to annotate the question x in the annotation bucket B_a with the few-shot ICL. To verify the correctness of teacher annotation t_e, we apply the similar TA-confidence as discussed in Section 3.1.2. The process of adding TA-confidence can be formulated as

$$t_e(x) = M_e(x, P_i), \quad s_e(x) = M_t(x, t_e, P_c)$$

where s_e is the TA-confidence of teacher annotations and serve as the TA-teacher signal. We calculate the confidence score for an example by

$$d(x) = \mathbb{1}_{s_e(x) \in C_3} \quad (2)$$

where C_3 are the confidence set. Teacher annotations will be added to the fine-tuning set B_f only if $d(x) \geq 1$. In our experiments, we set $C_3 = \{\text{very confident, confident}\}$. Note that our

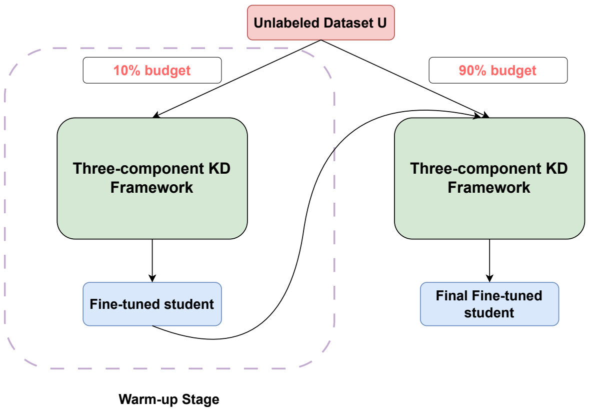

Figure 2: **Overview of two-stage training extension.** We first use 10% budget to fine-tune the student model and then use the other 90% to further training the student. Both stages use signals to enhance the fine-tuning.

signal does not require an additional cost to call the teacher model, which makes our approach different from (Madaan et al., 2023; Shinn et al., 2023) where they propose filtering out low-quality samples by self-refine from the teacher model. Examples of questions, annotations, and the TA signal s_e are shown in Table 15 in Appendix.

3.3 Fine-tune the Student Model

After collecting the fine-tuning dataset B_f, we fine-tune the student model with questions and teacher-generated annotations from the fine-tuning bucket. We apply the autoregressive language modeling as well as the cross entropy loss, which is the same as the pretraining objective to fine-tune the student model (Wang and Komatsuzaki, 2021).

3.4 Extension: Two-stage Training

While the aim of using student internal signal s_i is to actively select the examples not too simple or challenging for the student to learn, the student answers t for calculating s_i could be quite inaccurate, for example, the accuracy of using GPT-J-6B as the student is only 5.2% at HotpotQA task (Yang et al., 2018). With such a low accuracy, two challenges will emerge. First, the signal s_i cannot precisely reflect the difficulty and confidence of the student in the example. Second, the ratio of examples with $s_i(x) \in C_1$ is small (most examples with $s_i = 4/5$), causing a low sampling efficiency.

To address these challenges, as shown in Figure 2, we develop a two-stage training. The first stage is the warm-up training, and we use 10% of the budget to fine-tune the student with our framework (using signals). For the second stage with a more proficient student model, we use 90% of the budget to continue fine-tuning the student model with our framework. We expect that the student signals collected in the second stage are more reliable than the first stage, leading to a more performance boost.

Dataset	# Annotation Budget	# Test	Prompting Method	Metric	Task Type
HotpotQA	2,000	1,000	ReAct	Accuracy	Closed-book QA
GSM8K	2,000	1,319	CoT	Accuracy	Arithmetic reasoning
Aqua	2,000	254	CoT	Accuracy	Arithmetic reasoning
CSQA	2,000	1221	CoT	Accuracy	Commonsense reasoning

Table 2: Dataset statistics.

4 Experiments

Through our extensive empirical analysis, we aim to address the following research questions:

- **RQ1**: How efficient is our three-component framework compared to a classical KD method?
- **RQ2**: How important is each type of signal to the framework?
- **RQ3**: How well does our framework work for student and TA models of different sizes?
- **RQ4**: Does extending the framework to two-stage training boost more performance?

Dataset Since our work focuses on the scenario with sufficient unlabeled data but limited annotation budget, following the previous work (Wei et al., 2022; Yao et al., 2022), we evaluate our framework on four datasets with a large training set: HotpotQA (Yang et al., 2018), GSM8K (Cobbe et al., 2021), Aqua (Ling et al., 2017) and CommonSenseQA (Talmor et al., 2019). We choose HotpotQA, the closed-book question and answering (QA) dataset, for evaluation with ReAct prompting (Yao et al., 2022). For GSM8K and Aqua, the arithmetic reasoning task and CommonSenseQA, the commonse sense reasoning task, we evaluate performance with CoT prompting (Wei et al., 2022). In Table 2, we present the budget number, test number, prompting method, and evaluation metric of all four datasets.

Backbone models For the teacher model, we use gpt-3.5-turbo based on InstructGPT 175B (Ouyang et al., 2022) to generate the CoT or ReAct trajectories. We use Vicuna-13B and Vicuna-65B (Chiang et al., 2023) as TA models to evaluate the answers of the teachers and students. For the student models, we choose between the GPT-J-6B and Vicuna-13B models (Wang and Komatsuzaki, 2021).

Proposed framework We present two variants of the three-component framework with different signal weights as discussed in Section 3.1.3: In our first framework, we use $\alpha = 1$ and $\beta = 0$ (**TA-finetune (I)**); In our second framework (**TA-finetune (T)**, we use $\alpha = \beta = 1$ to compute annotation criteria. Both frameworks use the TA-teacher signal s_e as Eq (2). Throughout the experiments sections, we use "$\alpha = 1$" and "fine-tune with s_i", "$\beta = 1$" and "fine-tune with s_t" interchangeably.

Baseline models We design several baseline experiments to show the superiority of our proposed methods. The first group of baseline methods is the few-shot ICL methods: **Student-ICL**, **TA-ICL**, and **Teacher-ICL** with the corresponding prompting methods, that is, ReAct for the HotpotQA dataset and CoT for the others (Wei et al., 2022; Yao et al., 2022). The other baseline model is **Random-finetune**, the classical sequence-level KD pipeline for LLMs (Ho et al., 2022; Yao et al., 2022; He et al., 2023), where the examples are sampled randomly until the budget is met, and all annotations from the teacher model are used for student fine-tuning. Although there are other sequence-level KD methods, some depend on ground-truth labels (Li et al., 2023; Wang et al., 2023; Chen et al., 2023; Liu et al., 2023c; An et al., 2023; Zhao et al., 2023), not applicable in an unlabeled setting; those generate more augmented rationales (Liang et al., 2023; Shridhar et al., 2023; Feng et al., 2023) or more augmented questions (Yu et al., 2023), which are not achievable with a fixed budget; and those incorporate additional loss (Hsieh et al., 2023), which is orthogonal to our KD method. The details of the model configuration are included in Appendix A.

5 Results

5.1 RQ1: Comparison with Baseline Methods

Three-component KD framework outperforms Random-finetune and TA ICL We use GPT-J-6B as the student model and Vicuna-13B as the TA model. We experiment with two variants of the proposed framework: TA-finetune (I) and TA-finetune (T). As shown in Table 3, our proposed framework shows around 16.7% improvement compared to the Random-finetune baseline. It is also interesting to see that the fine-tuned GPT-J-6B model with the proposed framework can outperform the Vicuna-13B ICL model, which is already fine-tuned with 70k ChatGPT conversations (Chiang et al., 2023).

Method	HotpotQA	GSM8K	Aqua	CSQA	Avg.
Random	0.00	0.00	20.00	20.00	10.00
Teacher: GPT-3.5					
Few-shot ICL	27.54	73.50	52.55	75.02	57.15
TA: Vicuna-13B					
Few-shot ICL	17.57	18.80	22.00	60.98	29.84
Student: GPT-J-6B					
Few-shot ICL	5.20	4.20	21.30	21.91	13.15
Random-finetune	13.10	18.57	13.93	57.57	25.78
TA-finetune (I)	16.80	**19.33**	23.86	**60.36**	**30.09**
TA-finetune (T)	**17.70**	18.20	**26.77**	57.41	30.02

Table 3: **Performance of proposed three-component KD framework.** Accuracy (%) of teacher (GPT-3.5), TA (Vicuna-13B), student (GPT-J-6B) models on four datasets. "Random" refers to random-guess performance in multiple-choice tasks. The results of student fine-tuned models with our framework can consistently outperform random fine-tuning without any signals and even better than the ICL of TA models.

Note that the teacher ICL results are only 57.15% on average, which means that the proportion of noise in the teacher annotations cannot be easily ignored. Another example in point is the multiple-choice QA Aqua dataset, where the fine-tuned student model without any signal (13.93%) actually performs worse than its ICL counterpart (21.30%) as well as the random guess baseline (20.00%), and is primarily caused by the noise from teacher annotations. An example of noisy teacher annotations in the Aqua dataset - the teacher model generates rationales with "no answer" as the final output when it cannot find a satisfactory result among the given choices. The student model imitates the teacher's behavior and generates the similar "no answer" outputs, leading to a worse result than ICL. This case demonstrates the necessity of our annotation filtering signals to remove low-quality annotations. We also experiment with excluding the teacher annotations with a final "no answer" output and get 22.59%, and our methods with signals still achieve better performance.

5.2 RQ2: Ablation Study of Signals

After verifying the superiority of the overall framework on Random-finetune, we ask whether each component of signal presents its own functionality. In this section, we use the similar model and dataset setup in Section 5.1 and analyze the effects of each signal: TA-Teacher, TA-Student, and Student internal signal through ablation analysis.

Signal s_i	Signal s_e	HotpotQA	GSM8K	Aqua	CSQA	Avg.
N	N	13.10	18.57	13.93	**57.57**	25.78
N	Y	**16.20**	**18.87**	**25.51**	56.43	**29.25**
Y	N	13.80	**19.70**	15.91	59.13	27.14
Y	Y	**16.80**	19.33	**23.86**	**60.36**	**30.09**

(a) **Effects of TA-Teacher signal.** Results of fine-tuned students (GPT-J-6B) when considering our TA-Teacher signal. Signal s_t is not used for experiments in this table.

Signal s_i	Signal s_t	HotpotQA	GSM8K	Aqua	CSQA	Avg.
N	N	13.10	18.57	13.93	57.57	25.78
Y	N	13.80	19.70	**15.91**	59.13	**27.14**
N	Y	**14.20**	**20.30**	12.44	**59.71**	26.66

(b) **Effects of signals for annotation.** Results of fine-tuned student models (GPT-J-6B) on four datasets when adding or not TA-Student signals or student internal signals. Signal s_e is not used in the experiment of this table.

Table 4: Ablation study of all three signals. "Y" and "N" represent "yes" and "no" to indicate the signal use.

TA-teacher signal brings salient performance boost First, we compare the fine-tuned performance when adding or not the TA-Teacher signal in the condition of with or without the student internal signals s_i and show the results in Table 4a.

From Table 4a, we can observe that on average, excluding the uncertain annotations effectively boosts the performance, leading to the growth of the average accuracy from 25.78% to 29.25% or from 27.14% to 30.09% for fine-tuned with student disagreement (s_i) or not, respectively. Furthermore, the effects are more significant on more challenging datasets, such as HotpotQA and Aqua. This could be attributed to a significant improvement in data quality for these challenging datasets. We find that the ratio of the annotations with correct answer increases from 27.54% to 49.80% and from 52.55% to 75.57% for HotpotQA and Aqua datasets.

Signals for annotation show their effectiveness We separately demonstrate the effects of two types of signals for annotation (student internal s_i and TA-student signal s_t) using only one of them to actively select the questions to annotate. We summarize the results in Table 4b. The average results in Table 4b show that the use of either of the signals for annotation can increase performance from 25.78% to 27.14% or 26.66%, respectively, demonstrating the effectiveness of selecting uncertain samples to fine-tune the student model.

From the ablation study, we find that all types of signal have their own functionalities and orthogonal to each other. The performance of using only one type of signal can still exceed the results of the Random-finetune framework, and the combination of signals will lead to a more performance boost.

5.3 RQ3: Generalization on Students and TAs

The ablation analysis presents the effectiveness of each signal, and we examine whether the effect of the framework can be generalized to student or TA models of different sizes. For student models, we additionally use Vicuna-13B as a student model and compare with the results in Table 3 with GPT-J-6B as the student model. For TA models, we will show the effects with various TA models in two aspects: TA-teacher signal and TA-student signal.

5.3.1 Generalizations on Students Models

Student model	Signal s_i	Signal s_e	GSM8K	Aqua	CSQA	Avg.
GPT-J-6B	Y	Y	19.33	23.86	60.36	34.51
Vicuna-13B	N	N	38.06	16.14	71.41	41.87
Vicuna-13B	Y	N	39.58	19.37	**71.74**	43.56
Vicuna-13B	Y	Y	**40.71**	**29.92**	71.33	**47.32**

Table 5: **Effects of student model scale.** Results of fine-tuned student models with multiple scales. We apply Vicuna-13B and GPT-3.5 as TA and teacher models.To avoid the inductive bias of self-refinement of Vicuna-13B, we do not use TA-student signal s_t.

Signals are effective for students of various sizes We try Vicuna-13B as the student, and use another Vicuna-13B as the TA and GPT-3.5 as the teacher. We fine-tune Vicuna-13B as three signal combinations: no signals, only student internal signal s_i, and student internal s_i + TA-teacher signals s_e. We show the results of GPT-J-6B and Vicuna-13B as the student model, respectively, in Table 5. Note that fine-tuning Vicuna-13B model on HotpotQA dataset will cause out-of-memory issues due to the large length of ReAct trajectories, so we only report the fine-tuned performance on other datasets.

We can observe that in Table 5, fine-tuning with the Vicuna-13B student model leads to much better performance than the GPT-J-6B model, indicating that a larger model size will cause better fine-tuning results, which is consistent with previous findings in Ho et al. (2022). The best results for Vicuna-13B student models achieved with both TA-Student and TA-Teacher signals also verifies the generalizability of our framework on larger student models.

5.3.2 Generalizations on TA Models

TA models are used to examine both student and teacher answers. An intuitive question could be:

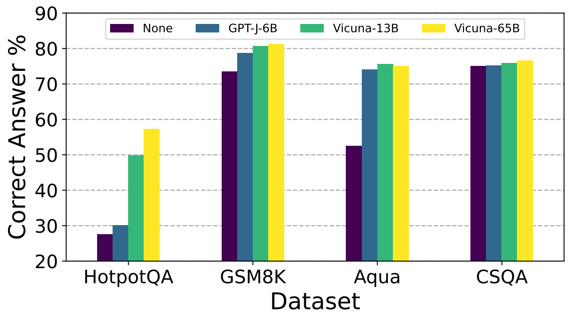

Figure 3: **Data quality of the fine-tuning bucket**. The proportion (%) of samples with correct annotations in the fine-tuning bucket when utilizing different scales of TA models in the TA-Teacher signal s_e.

How large could a model be an suitable choice of the TA? To explore this question, we use GPT-J-6B as the student, GPT-J-6B/Vicuna-13B/Vicuna-65B as the TA candidates, and GPT-3.5 as the teacher. To eliminate the other signals' influence, we experiment with only one signal: TA-teacher signal s_e or TA-student signal s_t in this part.

Choice of TA-Teacher signals Since the TA model in TA-teacher signal s_e filters incorrect teacher annotations to improve data quality in the fine-tuning bucket B_f, our hypothesis is that the data quality in B_f scales with the size of the TA, which influences the student fine-tuning result. To verify the hypothesis, we first collect the final teacher answer (the single answer) in different fine-tuning buckets. Then, we calculate the accuracy of the answer collection for each bucket, using the ground-truth answer from the original labeled dataset. We visualize the proportion of questions with correct teacher annotations with different TAs in Figure 3. Our observation is that with larger TAs, the fine-tuning data quality increases, but the improvement for model size from Vicuna 13B to 65B is smaller than that from GPT-J-6B to Vicuna-13B.

Then, we fine-tune the student model GPT-J-6B with the bucket B_f evaluated by different TA models. The results can be found in Table 6. From Table 6, we obtain the highest fine-tuning accuracy with the best TA, but the marginal improvement in data quality from 13B to 65B revealed in Figure 3 leads to comparable performance.

Choice of TA-student signals We vary the size of TA in the TA-student signals s_t to characterize the uncertainty of the student's inference. We use Vicuna-13B or Vicuna-65B as the TA to evaluate the student's inference from the GPT-J-6B model.

TA model	HotpotQA	GSM8K	Aqua	CSQA	Avg.
GPT-J-6B	13.20	17.43	23.15	52.74	26.63
Vicuna-13B	16.20	18.87	**25.51**	56.43	29.25
Vicuna-65B	**17.12**	**20.47**	24.41	**57.98**	**30.00**

Table 6: **Effects of TA model scale in the TA-Teacher signals.** Results of the fine-tuned student (GPT-J-6B) with different sizes of TAs in the TA-Teacher signals.

The fine-tuning results of the student model with only the TA-student signal are shown in Table 7.

TA model	HotpotQA	GSM8K	Aqua	CSQA	Avg.
Vicuna-13B	**14.20**	20.30	**12.44**	59.71	**26.66**
Vicuna-65B	13.40	**20.47**	12.20	**59.78**	26.46

Table 7: **Effects of TA model scale in the TA-student signals.** Results of the fine-tuned student (GPT-J-6B) with two different sizes of TAs in the TA-student signals.

From comparison with different TA models in the TA-student signal from Table 7, we can obtain comparable results when using a more advanced TA model to characterize the uncertainty of student's generations. We conjecture that the difference in difficulty between the annotation buckets B_a of different TA models, especially from 13B to 65B, is not that large, which may not show much significant effect on the final fine-tuning results.

Results in Tables 6 and 7 indicate a trade-off between the TA model sizes and the fine-tuning performance. Using a more advanced TA can improve some performance. However, in some scenarios, the computation resource can not afford a large TA model, such as the 65B model, then using the 13B model, which can effectively improve the fine-tuning data quality, is also a reasonable choice.

Moreover, we use different budget numbers to fine-tune the student model and present the detailed results in Appendix B.1. The observation from model performance can verify the generalization of our framework with various budget numbers.

5.4 RQ4: Effects of Two-stage Training

Stage	Signal s_i	Signal s_e	HotpotQA	GSM8K	Aqua	CSQA	Avg.
1	Y	N	**13.80**	19.70	21.26	**59.13**	28.47
2	Y	N	13.50	**20.17**	**23.46**	58.86	**29.00**
1	Y	Y	16.80	19.33	23.86	60.36	30.09
2	Y	Y	**16.90**	**21.30**	**25.98**	**60.36**	**31.14**

Table 8: **Effects of two-stage extension**. Results of fine-tuned student models (GPT-J-6B) on four datasets when extending from one-stage to two-stage training. Signal s_t is not used in the experiment of this table.

As we described in Section 3.4, we extend our one-stage training to two-stage training to better utilize the student internal signals. For the model and dataset setup, we use the same setting as in Section 5.1, which means that for the two-stage training, we use 200 examples for the first warm-up stage and the remaining 1,800 budgets for the second stage. For signal setup, we choose the student internal signal s_i and experiment under two conditions: with or without TA-Teacher signals s_e. We summarize the results in Table 8.

Two-stage extension obtain better results in most cases From the results in Table 8, we observe that extending the one-stage to the two-stage framework obtains a relative improvement of 1.86% and 3.48% for fine-tuning with or without TA-Teacher signals, respectively, and compared with the Random-finetune baseline, the two-stage framework brings the highest relative improvement of **20.79%**. Our extension obtains better results in 6/8 comparison cases, which verifies the effectiveness of the extension. The improvement can be attributed to more reliable signals s_i extracted by the student model in the second stage. We show the intermediate results in Appendix B.2 to interpret the improvement of the two-stage training. Moreover, we extend our fine-tuning to more stages to a curriculum learning pipeline, and the results of four-stage learning are presented in Appendix B.3.

6 Conclusions

In this paper, we propose a novel unified framework to resolve two challenges in the current teacher-student knowledge distillation process. Our framework utilizes a third TA model and signals from the student and teacher side to actively select samples and improve data quality in teacher annotations. Additionally, to better utilize student signals, we also extend our framework to two-stage training. With a limited budget for teacher annotations on an unlabeled dataset, our extensive empirical evaluations show that the proposed framework can significantly increase KD results with signals and all proposed signals show its own effectiveness.

7 Limitations

In our framework, we apply several types of signals: student intenal signal, TA-student and TA-teacher signal to refine the knowledge distillation of LLM.

In addition to these internal and external signals, we can try other signals developed in existing works about active learning (Zhang et al., 2022; Settles, 2009; Li et al., 2024a). For example, we can aggregate the model confidence on each generated token in the rationales as the student or teacher's confidence on their annotations. Moreover, in addition to querying the TA model of confidence on the final answers, we can also ask the TA to evaluate the intermediate reasoning steps of the teacher or student models on each question, which could be another type of external signal.

We have verified the effectiveness of our framework on multiple models, prompting methods, and datasets. In related work, we have disccused multiple sequence-level KD methods, using more advanced fine-tuning loss (Hsieh et al., 2023) and more complex generated rationales (Shridhar et al., 2023) to improve KD performance. These methods are complementary to our three-component framework, aiming to increase the results with sufficiently large teacher-annotated data. Our method could further enhance their efficiency in fixed-budget and unlabeled data settings. We leave the exploration of the combination with existing works to the future.

The student models (GPT-J-6B and Vicuna-13B) in the experiments are fine-tuned on 8 NVIDIA A100 GPUs, which may not be accessible to everyone and have a negative environmental impact. For further experiments, we can extend our experiments to smaller language models with fewer than 1 billion parameters as the student model. Furthermore, we can also use more advanced teacher models, such as GPT-4 (OpenAI, 2023), to observe whether a better teacher model leads to a greater improvement in performance.

Acknowledgments

We would like to thank the anonymous reviewers as well as Paiheng Xu for reviewing the article and for providing helpful comments and suggestions.

References

Shengnan An, Zexiong Ma, Zeqi Lin, Nanning Zheng, Jian-Guang Lou, and Weizhu Chen. 2023. Learning from mistakes makes llm better reasoner. *arXiv preprint arXiv:2310.20689*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Mizil. 2006. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 535–541.

Hongzhan Chen, Siyue Wu, Xiaojun Quan, Rui Wang, Ming Yan, and Ji Zhang. 2023. Mcc-kd: Multi-cot consistent knowledge distillation. *arXiv preprint arXiv:2310.14747*.

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

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

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

Antonia Creswell and Murray Shanahan. 2022. Faithful reasoning using large language models. *arXiv preprint arXiv:2208.14271*.

Antonia Creswell, Murray Shanahan, and Irina Higgins. 2022. Selection-inference: Exploiting large language models for interpretable logical reasoning. *arXiv preprint arXiv:2205.09712*.

Shizhe Diao, Pengcheng Wang, Yong Lin, and Tong Zhang. 2023. Active prompting with chain-of-thought for large language models. *arXiv preprint arXiv:2302.12246*.

Tao Feng, Zifeng Wang, and Jimeng Sun. 2023. Citing: Large language models create curriculum for instruction tuning. *arXiv preprint arXiv:2310.02527*.

Nan He, Hanyu Lai, Chenyang Zhao, Zirui Cheng, Junting Pan, Ruoyu Qin, Ruofan Lu, Rui Lu, Yunchen Zhang, Gangming Zhao, et al. 2023. Teacherlm: Teaching to fish rather than giving the fish, language modeling likewise. *arXiv preprint arXiv:2310.19019*.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Namgyu Ho, Laura Schmid, and Se-Young Yun. 2022. Large language models are reasoning teachers. *arXiv preprint arXiv:2212.10071*.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alexander Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. 2023. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. *arXiv preprint arXiv:2305.02301*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Yiwei Li, Peiwen Yuan, Shaoxiong Feng, Boyuan Pan, Bin Sun, Xinglin Wang, Heda Wang, and Kan Li. 2023. Turning dust into gold: Distilling complex reasoning capabilities from llms by leveraging negative data. *arXiv preprint arXiv:2312.12832*.

Zongxia Li, Andrew Mao, Daniel Stephens, Pranav Goel, Emily Walpole, Alden Dima, Juan Fung, and Jordan Boyd-Graber. 2024a. Improving the TENOR of labeling: Re-evaluating topic models for content analysis. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 840–859, St. Julian's, Malta. Association for Computational Linguistics.

Zongxia Li, Ishani Mondal, Yijun Liang, Huy Nghiem, and Jordan Lee Boyd-Graber. 2024b. Panda (pedantic answer-correctness determination and adjudication):improving automatic evaluation for question answering and text generation.

Zhenwen Liang, Dian Yu, Xiaoman Pan, Wenlin Yao, Qingkai Zeng, Xiangliang Zhang, and Dong Yu. 2023. Mint: Boosting generalization in mathematical reasoning via multi-view fine-tuning. *arXiv preprint arXiv:2307.07951*.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. Program induction by rationale generation: Learning to solve and explain algebraic word problems. *arXiv preprint arXiv:1705.04146*.

Fuxiao Liu, Kevin Lin, Linjie Li, Jianfeng Wang, Yaser Yacoob, and Lijuan Wang. 2023a. Aligning large multi-modal model with robust instruction tuning. *arXiv preprint arXiv:2306.14565*.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023b. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35.

Weize Liu, Guocong Li, Kai Zhang, Bang Du, Qiyuan Chen, Xuming Hu, Hongxia Xu, Jintai Chen, and Jian Wu. 2023c. Mind's mirror: Distilling self-evaluation capability and comprehensive thinking from large language models. *arXiv preprint arXiv:2311.09214*.

Xiaoyu Liu, Paiheng Xu, Junda Wu, Jiaxin Yuan, Yifan Yang, Yuhang Zhou, Fuxiao Liu, Tianrui Guan, Haoliang Wang, Tong Yu, et al. 2024. Large language models and causal inference in collaboration: A comprehensive survey. *arXiv preprint arXiv:2403.09606*.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. 2023. Self-refine: Iterative refinement with self-feedback. *arXiv preprint arXiv:2303.17651*.

Seyed Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghasemzadeh. 2020. Improved knowledge distillation via teacher assistant. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 5191–5198.

OpenAI. 2023. Gpt-4 technical report.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Baolin Peng, Michel Galley, Pengcheng He, Hao Cheng, Yujia Xie, Yu Hu, Qiuyuan Huang, Lars Liden, Zhou Yu, Weizhu Chen, et al. 2023. Check your facts and try again: Improving large language models with external knowledge and automated feedback. *arXiv preprint arXiv:2302.12813*.

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.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Timo Schick and Hinrich Schütze. 2020. Exploiting cloze questions for few shot text classification and natural language inference. *arXiv preprint arXiv:2001.07676*.

Timo Schick and Hinrich Schütze. 2021. It's not just size that matters: Small language models are also few-shot learners. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2339–2352, Online. Association for Computational Linguistics.

Burr Settles. 2009. Active learning literature survey.

Noah Shinn, Beck Labash, and Ashwin Gopinath. 2023. Reflexion: an autonomous agent with dynamic memory and self-reflection. *arXiv preprint arXiv:2303.11366*.

Kumar Shridhar, Alessandro Stolfo, and Mrinmaya Sachan. 2023. Distilling reasoning capabilities into smaller language models. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 7059–7073.

Chenglei Si, Zhe Gan, Zhengyuan Yang, Shuohang Wang, Jianfeng Wang, Jordan Boyd-Graber, and Lijuan Wang. 2022. Prompting gpt-3 to be reliable. *arXiv preprint arXiv:2210.09150*.

Wonchul Son, Jaemin Na, Junyong Choi, and Wonjun Hwang. 2021. Densely guided knowledge distillation using multiple teacher assistants. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9395–9404.

Hongjin Su, Jungo Kasai, Chen Henry Wu, Weijia Shi, Tianlu Wang, Jiayi Xin, Rui Zhang, Mari Ostendorf, Luke Zettlemoyer, Noah A Smith, et al. 2022. Selective annotation makes language models better few-shot learners. *arXiv preprint arXiv:2209.01975*.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`.

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

Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. `https://github.com/kingoflolz/mesh-transformer-jax`.

Peiyi Wang, Lei Li, Liang Chen, Feifan Song, Binghuai Lin, Yunbo Cao, Tianyu Liu, and Zhifang Sui. 2023. Making large language models better reasoners with alignment. *arXiv preprint arXiv:2309.02144*.

Xiyao Wang, Jiuhai Chen, Zhaoyang Wang, Yuhang Zhou, Yiyang Zhou, Huaxiu Yao, Tianyi Zhou, Tom Goldstein, Parminder Bhatia, Furong Huang, and Cao Xiao. 2024a. Enhancing visual-language modality alignment in large vision language models via self-improvement.

Xiyao Wang, Yuhang Zhou, Xiaoyu Liu, Hongjin Lu, Yuancheng Xu, Feihong He, Jaehong Yoon, Taixi Lu, Gedas Bertasius, Mohit Bansal, et al. 2024b. Mementos: A comprehensive benchmark for multimodal large language model reasoning over image sequences. *arXiv preprint arXiv:2401.10529*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

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

Sean Welleck, Ximing Lu, Peter West, Faeze Brahman, Tianxiao Shen, Daniel Khashabi, and Yejin Choi. 2022. Generating sequences by learning to self-correct. *arXiv preprint arXiv:2211.00053*.

Benfeng Xu, Licheng Zhang, Zhendong Mao, Quan Wang, Hongtao Xie, and Yongdong Zhang. 2020. Curriculum learning for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6095–6104.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*.

Kang Min Yoo, Dongju Park, Jaewook Kang, Sang-Woo Lee, and Woomyeong Park. 2021. Gpt3mix: Leveraging large-scale language models for text augmentation. *arXiv preprint arXiv:2104.08826*.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. 2023. Metamath: Bootstrap your own mathematical questions for large language models. *arXiv preprint arXiv:2309.12284*.

Zhisong Zhang, Emma Strubell, and Eduard Hovy. 2022. A survey of active learning for natural language processing. *arXiv preprint arXiv:2210.10109*.

Jiachen Zhao, Wenlong Zhao, Andrew Drozdov, Benjamin Rozonoyer, Md Arafat Sultan, Jay-Yoon Lee, Mohit Iyyer, and Andrew McCallum. 2023. Multi-stage collaborative knowledge distillation from large language models. *arXiv preprint arXiv:2311.08640*.

Yiyang Zhou, Zhiyuan Fan, Dongjie Cheng, Sihan Yang, Zhaorun Chen, Chenhang Cui, Xiyao Wang, Yun Li, Linjun Zhang, and Huaxiu Yao. 2024a. Calibrated self-rewarding vision language models.

Yuhang Zhou, Suraj Maharjan, and Beiye Liu. 2023. Scalable prompt generation for semi-supervised learning with language models. *arXiv preprint arXiv:2302.09236*.

Yuhang Zhou, Paiheng Xu, Xiaoyu Liu, Bang An, Wei Ai, and Furong Huang. 2024b. Explore spurious correlations at the concept level in language models for text classification.

A Implementation Details

A.1 Signal and annotation generation

For all inference trajectories of the student models for student internal signal, TA-student signal and test inference, following the previous work, we set the maximum sequence length to 1,024 (Ho et al., 2022; Kojima et al., 2022). For teacher models' annotations, we set the maximum sequence length to 2,048 for GPT-3.5.

We utilize greedy search in decoding for all generations, except for the students' generations for the collection of student internal signals s_i, where we use stochastic temperature sampling with the same temperature value 0.7 as in the previous work (Diao et al., 2023; Ho et al., 2022; Wang et al., 2022; Zhou et al., 2024b,a; Liu et al., 2024).

We use the same few-shot ICL prompts as in previous work to generate student or teacher annotations for different datasets (Wei et al., 2022; Wang et al., 2024a; Diao et al., 2023; Wang et al., 2024b) and for the crafted confidence prompt P_c, the details and exemplars inside the prompts are shown in Tables 12 and 13.

We call the gpt-3.5-turbo function from OpenAI to generate teacher annotations and rationales. The price of this API is $0.0015 / 1K tokens for inputs and $0.002 / 1K tokens for output. The total expenditure on API usage is $ 207.05, including preliminary exploration.

A.2 Student model fine-tuning

For the fine-tuning of the student model, we base our implementation on the Pytorch[1], Huggingface transformer[2], and Stanford Alpaca[3]. We use AdamW as our optimizer with a learning rate of

[1] `https://pytorch.org/`
[2] `https://huggingface.co/`
[3] `https://github.com/tatsu-lab/stanford_alpaca`

2e−5 and a weight decay of 0.01 with linear scheduler, batch size of 1, and trained for 5 epochs with the early stopping mechanism.

B Supplementary Experimental Results

In this section, we will show the supplementary experimental results, including our framework with different budgets and the performance of extending our framework to curriculum learning.

B.1 Generalizations on Different Budgets

In Section 5.1, we have verified the effectiveness of our framework with 2,000 as the budget number, and to show the generalization of our framework on different budget numbers, we will repeat the experiments with the same model and dataset setup with the budget numbers 1,000 and 3,000, respectively. We present the results in Table 9.

Method	HotpotQA	GSM8K	Aqua	CSQA	Avg.
		Budget: 1000			
Random-finetune	10.20	**15.46**	10.08	53.97	22.43
TA-finetune (I)	**15.90**	13.94	20.16	**59.46**	**27.37**
TA-finetune (T)	13.70	13.12	**23.54**	57.41	26.94
		Budget: 3000			
Random-finetune	13.40	21.61	13.93	57.57	26.63
TA-finetune (I)	**17.10**	21.30	21.02	**59.46**	29.72
TA-finetune (T)	16.20	**22.51**	**24.17**	57.75	**30.16**

Table 9: **Effects of budget number.** Results of fine-tuned student models (GPT-J-6B) on four datasets when setting the budget number to 1,000 or 3,000 respectively.

From Tables 9 and 3, we find that the performance of all methods improves a lot when the budget number increases from 1,000 to 2,000 and will get a comparable performance when the budget increases from 2,000 to 3,000. Our proposed three-component KD framework can outperform the random-finetune method in all three cases, which suggests that our framework's effectiveness can be generalized to different numbers of budgets.

B.2 Details of Two-stage Training

For the interpretation of the improvement in two-stage training in Section 5.4, we conjecture that it could be attributed to two reasons. First, after the warm-up stage, the better student model is more expert in the questions, so the student's internal signal s_i should be more reliable. Second, samples extracted by student internal signals s_i (the uncertain questions for the student) should be more challenging in the second stage and continually fine-tuning the student with increasingly difficult data, leading to a deeper learning curve and a higher performance boost. To verify these two hypotheses, we present the accuracy of the student model and the difficulty level of the samples in the annotation bucket B_a before and after the warm-up stage.

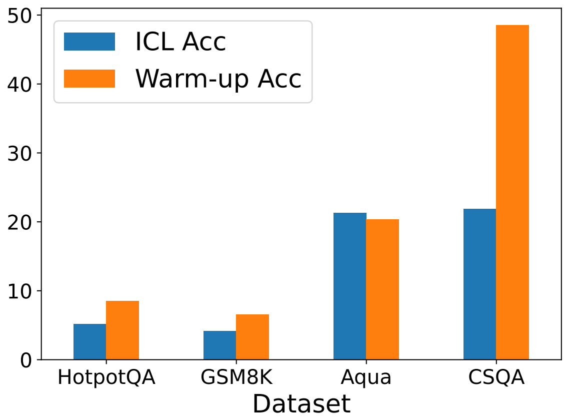

Figure 4: **Accuracy comparison of the student in the first and second stage.** The student ICL accuracy (ICL Acc) in the first stage and the student accuracy in the second stage (Warm-up Acc). After fine-tuning on a small dataset to warm up, the student model becomes more promising and expert in answering complex questions in most cases.

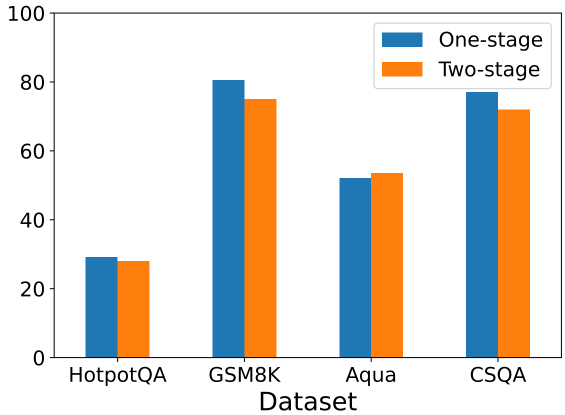

Figure 5: **Data difficulty in the annotation bucket.** Teacher model's accuracy on the annotation bucket B_a in the first stage and second stage, respectively. The teacher model's accuracy on the questions in B_a for the second stage is lower than that for the first stage and suggests more difficult and challenging questions in B_a.

We first present the student ICL accuracy (ICL Acc) in the first stage and the student accuracy in the second stage (Warm-up Acc) in Figure 4. We find that after fine-tuning on a small dataset to warm up, the student model becomes more promising and expert in answering complex questions in most cases.

Then, we also quantify the difficulty of samples in the annotation bucket B_a of in the first stage and second stage, respectively, by measuring the teacher model accuracy on questions in B_a. We visualize the accuracy of the teacher model of the samples in the annotation bucket in two stages in Figure 5. We find that in most datasets, the teacher model's accuracy on the questions in the annotation bucket for the second stage is lower than that for the first stage and suggests more difficult and challenging questions in B_a. This observation indicates that continually fine-tuning the student model on the slightly more difficult data causes a deeper learning curve, leading to a better performance.

B.3 Explorations on Curriculum Learning

Inspired by the results of the two-stage training, our next question is whether the performance improvement will continue to increase when using more stages. We can frame multiple-stage training into the curriculum learning scope, which means that at each stage, we will have better student models and the samples extracted by student signals should be more difficult and challenging (Xu et al., 2020). For our experiment setup, we divide our training into four stages and use 200, 400, 600, and 800 budgets (a total of 2,000) in each stage. For the final step, we will integrate all the annotations in every stage and continue fine-tuning the student model for a few epochs. We present the results of one-stage training and curriculum learning in Table 10.

Training Method	Signal s_i	Signal s_t	HotpotQA	GSM8K	Aqua	Avg.
One	Y	N	**13.80**	**19.70**	**15.91**	**16.47**
CL	Y	N	11.21	19.60	14.96	15.26
One	Y	Y	**16.80**	**19.33**	**23.86**	**19.99**
CL	Y	Y	16.60	18.57	22.52	19.23

Table 10: **Effects of curriculum learning**. Results of fine-tuned student models (GPT-J-6B) on three datasets when extending the one-stage training to a curriculum learning method (four-stage training).

From Table 10, we do not observe improvement as in the results of the two-stage training, and we conjecture that there should exist multiple reasons to influence the results of curriculum learning, such as the learning rate for each stage and the number of budgets in each stage. Here, we keep the learning rate for each stage as the same value, which may cause the fine-tuning to a sub-optimal point. We leave the further exploration of curriculum learning of our framework as a future direction.

B.4 Preliminary Explorations of Annotation Criteria

In Section 3.1.3, we choose $C_1 = \{2, 3\}$ as the annotation criteria of the internal signal s_i, and this hyperparameter choice is decided by our preliminary experiment. We compare the fine-tuning result with only s_i (without signal s_t and s_e) by choosing $C_1 = \{2, 3\}$ and $C'_1 = \{4, 5\}$ and without s_i (randomly sampled) by fixing budget number 300 in the HotpotQA dataset. The model configuration is the same as in Section 5.1. The result is shown in Table 11.

Task	Few-shot ICL	Finetuned without s_i	Finetuned with C'_1	Finetuned with C_1
HotpotQA	5.20%	8.50%	7.50%	**10.50%**

Table 11: **Preliminary exploration of annotation criteria on HotpotQA dataset.** Results of the fine-tuned student model (GPT-J-6B) on HotpotQA when fine-tuned without s_i (randomly sampled questions), with s_i and $C_1 = \{2, 3\}$ and $C'_1 = \{4, 5\}$.

From Table 11, we find that the student model fine-tuned with $C_1 = \{2, 3\}$ achieves the best performance, indicating that fine-tuning with $C_1 = \{2, 3\}$ can bring the deepest learning curve, so we choose this hyperparameter choice in the following experiments. Furthermore, we observe that the model fine-tuned with the annotation criteria C'_1 cannot outperform random selection without signal s_i, suggesting that it is hard for the student to learn useful information from hard questions.

You are an advanced reasoning agent. You will be given a previous reasoning trial in which you were given access to a Docstore API environment and the answer to the question. Report the confidence about the answer from choices (a) very confident (b) confident (c) not confident (d) wrong answer. Here are some examples: {demonstrations} Previous Trial: {model generations} Confidence Choice:

(a) Prompt P_c for HotpotQA dataset.

You are an advanced {question type} agent. You will be given a previous {question type} question, the reasoning trail and the answer to the question. Report the confidence about the answer from choices (a) very confident (b) confident (c) not confident (d) wrong answer. Here are some examples: {demonstrations} Previous Trial: {model generations} Confidence Choice:

(b) Prompt P_c for GSM8K, Aqua and CommonsenseQA dataset. For GSM8K and Aqua, {question type} is "arithmetic reasoning" and for CommonsenseQA, it is "commonsense reasoning".

Table 12: Prompt P_c for all datasets. {demonstrations} is a placeholder to add four demonstrations with provided ground-truth TA confidence for each dataset. {model generations} is the place to insert the teacher or student generated annotations.

Q: The Fiery Furnaces and Paul Westerberg, are of which nationality?
Thought 1: I need to search The Fiery Furnaces and Paul Westerberg, find their nationality, and answer the question.
Action 1: Search[The Fiery Furnaces]
Observation 1: [truncated wikipedia excerpt]
Thought 2: The Fiery Furnaces is an American band. I need to search Paul Westerberg next and find his nationality.
Action 2: Search[Paul Westerberg]
Observation 2: Paul Harold Westerberg (born December 31, 1959) is an American musician, best known as the lead singer, guitarist, and songwriter for The Replacements.[2] Following the breakup of The Replacements, Westerberg launched a solo career that saw him release three albums on two major record labels.. Following the release of his third solo album, Westerberg has been mostly releasing music that he has self-produced and recorded in his basement home studio.[3] He has also released two albums and an EP under the pseudonym Grandpaboy. In 2017, Westerberg released songs on SoundCloud as User 964848511 and on Bandcamp as Dry Wood Garage.[4][5][6]. In the late 1970s, Westerberg was working as a janitor for U.S. Senator David Durenberger,[7] and one day while walking home from work, he heard a band practicing Yes's "Roundabout" in a basement.
Thought 3: Paul Westerberg is also an American musician. So The Fiery Furnaces and Paul Westerberg are both American.
Action 3: Finish[American]
Confidence Choice: (a) very confident

(a) One of the examples in the demonstrations of the Prompt P_c for TA confidence generation for HotpotQA dataset

Q: Abraham owns 80 square meters of unused land. He sold half of the land for $50, and after a month, he sold another 1/4 of his land for $30. He then sold the remaining land for $3 per square meter. How much money will he be able to earn after selling all his unused land?
A: Let's think step by step. Abraham owns 80 square meters of unused land. He sold half of the land for $50. After a month, he sold another 1/4 of his land for $30. He then sold the remaining land for $3 per square meter. 80 - 30 = 50. Then he sold the remaining land for $3 per square meter. 50 - 3 = 47. The answer is 47.
Confidence Choice: (d) wrong answer

(b) One of the examples in the demonstrations of the Prompt P_c for TA confidence generation for GSM8K dataset

Q: Alfred buys an old scooter for $4700 and spends $800 on its repairs. If he sells the scooter for $5800, his gain percent is? Answer Choices: (A) 6.23% (B) 5.45% (C) 7% (D) 8.12% (E) 10%
A: Let's think step by step. The total cost of the scooter for Alfred is $4700 + $800 = $5500. His gain is the selling price minus the cost price, which is $5800 - $5500 = $300. The gain percent is (gain / cost price) * 100% = ($300 / $5500) * 100% = 5.45% (rounded to two decimal places). The answer is B.
Confidence Choice: (a) very confident

(c) One of the examples in the demonstrations of the Prompt P_c for TA confidence generation for Aqua dataset

Q: When old conservative people observe a changing society, what do the experience? Answer Choices: (A) depression (B) resentment (C) hurt (D) anger (E) acceptance
A: Let's think step by step. The answer should be the feeling that old conservative people experience when observing a changing society. Of the above choices, the closest feeling is resentment. The answer is B.
Confidence Choice: (c) not confident

(d) One of the examples in the demonstrations of the Prompt P_c for TA confidence generation for CommonsenseQA dataset

Table 13: Examples in the demonstrations of the Prompt P_c for TA confidence (TA-student signal s_t and TA-teacher signal s_e) generation for all four datasets

Question	On Friday, Markeesha sold 30 boxes of crackers for her scout troop's fundraiser. On Saturday, she sold twice as many as on Friday. On Sunday, she sold 15 fewer than Saturday. How many boxes did she sell over the three days?
Generation 1	Let's think step by step. Markeesha started with 30 boxes of crackers. On Friday, she sold 30 boxes. On Saturday, she sold 70 boxes. On Sunday, she sold 20 less than Saturday. So she sold 60 boxes - 20 = 40 boxes. The answer is 40.
Generation 2	Let's think step by step. Markeesha started with 30 boxes. On friday, she sold 30 boxes. On saturday, she sold 60 boxes. On saturday, she sold 60 boxes - 15 = 45. The answer is 45.
Generation 3	Let's think step by step. Markeesha started with 30 boxes. On Friday, she sold 30 boxes for her troop's fundraiser. On Saturday, she sold 60 boxes for her troop's fundraiser. On Sunday, she sold 15 fewer than Saturday. So she sold 30 + 60 - 15 = 45 boxes. The answer is 45.
Generation 4	Let's think step by step. On Friday, she sold 30 boxes. On Saturday, she sold twice as many. So she sold 60 boxes. On Sunday, she sold 15 fewer than Saturday. So she sold 45 boxes. The answer is 45.
Generation 5	Let's think step by step. Markeesha sold 30 boxes of crackers on Friday. On Saturday, she sold twice as many as on Friday. So she sold 2 * 30 = 60 boxes. On Sunday, she sold 15 fewer than Saturday. So she sold 60 - 15 = 45 boxes. The answer is 45.

(a) Examples of the question with student disagreement = 2 in GSM8k dataset

Question	There are 30 pieces of popcorn in a serving. Jared can eat 90 pieces of popcorn and his three other friends can each eat 60 pieces of popcorn. How many servings of popcorn should Jared order for all of them?
Generation 1	Let's think step by step. There are 30 pieces of popcorn in the serving. Jared can eat 90 pieces of popcorn. His three other friends can each eat 60 pieces of popcorn. So Jared needs to order 3 * 90 = 270 pieces of popcorn. The answer is 270.
Generation 2	Let's think step by step. There are initially 30 pieces of popcorn in a serving. Jared can eat 90 pieces of popcorn and his three other friends can each eat 60 pieces of popcorn. So 90 + 60 = 150. Jared should order 150 pieces. Jared can eat 150 - 30 = 120 pieces of popcorn. His three friends can eat 60 pieces of popcorn. So Jared's three friends can each eat 90 - 60 = 30 pieces of popcorn. The answer is 30.
Generation 3	Let's think step by step. Jared started with 30 pieces of popcorn, but he can eat 90 pieces of popcorn. 60 pieces each is 3 x 30 = 90 pieces. Jared has 90 - 60 pieces. Jared should order 30 servings of popcorn for all of the friends. The answer is 30.
Generation 4	Let's think step by step. There are 30 pieces of popcorn. Jared can eat 90 pieces. The other three can eat 60 pieces. So Jared needs 90 + 60 = 150 pieces of popcorn. The answer is 150.
Generation 5	Let's think step by step. Jared can eat 90 pieces. So he can eat 30 pieces a serving. His three friends can each eat 60 pieces. 60 pieces * 3 = 180 pieces. 30 pieces x 180 = 5400 pieces. Jared can order 5400 pieces for all of them. The answer is 5400.

(b) Examples of the question with student disagreement = 4 in GSM8k dataset

Table 14: Examples of questions as well as the student models' generations with different disagreement (student internal signal s_i).

Question	There are 290 liters of oil in 24 cans. If 10 of the cans are holding 8 liters each, how much oil is each of the remaining cans holding?
Teacher Generation	Let's think step by step. There are 290 liters of oil in 24 cans. If 10 of the cans are holding 8 liters each, then the total amount of oil in those 10 cans is 10 x 8 = 80 liters. So the remaining oil in the other cans is 290 - 80 = 210 liters. Since there are 24 - 10 = 14 remaining cans, each of those cans is holding 210 / 14 = 15 liters. Therefore, each of the remaining cans is holding 15 liters of oil.
TA Confidence	Confidence Choice: (a) very confident

(a) Examples of the question, teacher annotation and TA confidence = "very confident" in GSM8k dataset

Question	It is recommended that men should consume no more than 150 calories of added sugar per day. Mark took a soft drink in the afternoon that contained 2500 calories, 5% of which was from added sugar. Then he bought some bars of candy which had 25 calories of added sugar each. If he eventually exceeded the recommended intake of added sugar by 100%, how many bars of candy did he take?
Teacher Generation	Let's think step by step. First, we need to find out how many calories of added sugar Mark consumed from the soft drink. Since 5% of the soft drink's calories were from added sugar, we can calculate 2500 * 0.05 = 125 calories of added sugar from the soft drink.
TA Confidence	Confidence Choice: (d) wrong answer

(b) Examples of the question, teacher annotation and TA confidence = "wrong answer" in GSM8k dataset

Table 15: Examples of questions as well as the teacher models' generations with the corresponding TA confidence (TA-teacher signal s_e).