Scaling LLM Reasoning from Minimal Labels: A Semi-Supervised Framework with a Lightweight Verifier

Keizo Kato[1,2], Chenhui Chu[2], Yugo Murawaki[2], Sado Kurohashi[2,3]
[1]Fujitsu Limited, Kawasaki, Japan
[2]Kyoto University, Kyoto, Japan
[3]National Institute of Informatics, Tokyo, Japan
kato.keizo@jp.fujitsu.com, {chu, murawaki, kuro}@i.kyoto-u.ac.jp

Abstract

For the development of Large language models (LLMs), recent approaches to generating pseudo intermediate reasoning have shown remarkable progress. But they typically rely on large numbers of correctly annotated answers to assess reasoning quality. This paper presents a semi-supervised framework that scales reasoning learning from minimal supervision, turning reasoning verification itself into a data creation mechanism. We train a lightweight reasoning-correctness classifier on only a few labeled samples, which judges whether intermediate reasoning traces generated by an LLM are valid. Furthermore, an entropy-based confidence threshold filters out unreliable samples, and the remaining high-confidence reasoning traces are used to fine-tune the model. Experiments on Verifiable Math Problems (Orca-Math subset) and Question Answering on Image Scene Graphs (GQA) with Visual Programming show that our method achieves accuracy comparable to using 10–15× more labeled data. Ablation analyses confirm that both the classifier and entropy filtering are essential for scalable and noise-resistant pseudo-labeling. By replacing expensive answer-level supervision with lightweight reasoning verification, our method provides a practical path toward constructing large-scale reasoning resources and paves the way for future autonomous reasoning systems that learn from minimal human input.

Keywords: semi-supervised learning, (Semi-)Automatic Generation of Training Data, pseudo-label

1. Introduction

Recent advances in Large Language Models (LLMs) have been driven not only by scaling laws but also by the explicit modeling of *intermediate reasoning* processes. Such reasoning traces, often referred to as chains of thought, allow models to decompose complex problems into interpretable substeps, thereby improving both accuracy and robustness across various tasks. A series of studies—including Chain-of-Thought prompting (Wei et al., 2022), Self-Consistency (Wang et al., 2023), Tree-of-Thoughts (Yao et al., 2023a), and ReAct (Yao et al., 2023b)—have demonstrated that eliciting intermediate reasoning from LLMs significantly enhances performance on tasks requiring multi-step inference and logical composition.

However, constructing large-scale datasets containing intermediate reasoning annotations is prohibitively expensive and time-consuming. To address this challenge, a growing body of work has explored *pseudo reasoning generation*, in which LLMs themselves generate intermediate reasoning traces that are then used for self-training. Methods such as Self-Training (Zelikman et al., 2022), Reflexion (Shinn et al., 2023), and Meta-CoT (Xiang et al., 2025) adopt the assumption that if an LLM's final answer is correct, then its reasoning process leading to that answer can also be regarded as correct. This paradigm enables the automatic creation of reasoning-augmented data from question–answer pairs, alleviating the need for human-written explanations.

Nevertheless, these approaches still rely on the availability of *answer annotations*. In many real-world domains—such as education, law, or medicine—obtaining correct answers requires expert verification and incurs substantial annotation costs. By contrast, *problem statements themselves* are often abundant and inexpensive to collect, for example from textbooks, public Question and Answering (QA) corpora, or web-based question repositories. Thus, while questions are plentiful, answer labels remain a scarce and costly resource.

In this paper, we propose a novel training framework that enhances LLM performance even when only a few answer-annotated samples are available, leveraging a large pool of unlabeled question data. Specifically, we first train a **binary classifier** that determines whether an LLM-generated reasoning trace and its corresponding answer are correct, using only a small set of labeled examples. This classifier is then applied to unlabeled questions: the LLM generates intermediate reasoning and candidate answers, which are filtered by the classifier, and only those judged as "correct" are adopted as pseudo-labeled data. Furthermore, by introducing an **entropy-based confidence threshold**, we dynamically select reliable samples, leading to substantial performance gains with minimal supervision.

Our main contributions are threefold:

Proceedings of the Fifteenth Language Resources and Evaluation Conference (LREC 2026), pages 6155–6165
11–16 May 2026.

1. We propose a learning framework that requires only a small number of answer-annotated samples while exploiting large-scale unlabeled question data to expand reasoning supervision.
2. We introduce a classifier that judges the correctness of LLM-generated reasoning traces, enabling high-quality pseudo reasoning data selection.
3. We demonstrate that entropy-based filtering substantially improves both accuracy and data efficiency, outperforming existing self-training methods in low-supervision scenarios.

2. Related Work

Our study relates to reasoning enhancement with intermediate thoughts, pseudo-labeling for self-training, confidence-based sample selection, and self-improvement frameworks.

2.1. Reasoning Enhancement via Intermediate Thoughts

A growing body of research has shown that encouraging LLMs to produce explicit reasoning steps improves their ability to solve complex problems. Chain-of-Thought (CoT) prompting (Wei et al., 2022) demonstrated that simple prompt engineering can elicit step-by-step reasoning. Self-Consistency (Wang et al., 2023) enhanced reasoning stability by sampling multiple reasoning traces and aggregating them through majority voting. Tree-of-Thoughts (Yao et al., 2023a) proposed a search-based reasoning framework that evaluates alternative reasoning branches systematically, while ReAct (Yao et al., 2023b) and Reflexion (Shinn et al., 2023) integrate reasoning, action, and self-correction for improved coherence. Our work follows this line of research but focuses on treating reasoning traces as data resources whose quality can be automatically filtered and reused.

2.2. Pseudo-Label Generation and Self-Training

Since manually annotating reasoning steps is costly, several studies have explored generating and reusing pseudo reasoning via self-training. Self-Training for reasoning (Zelikman et al., 2022) proposed iterative fine-tuning on model-generated explanations, while Meta-CoT (Xiang et al., 2025) introduced a meta-learning framework to recursively improve reasoning generation. Most of these methods assume that "if the final answer is correct, the reasoning is also correct." Our approach relaxes this assumption by introducing a learned classifier that discriminates between correct and incorrect reasoning, thereby improving the robustness of pseudo-label selection. In addition to correctness-based pseudo-labeling, recent self-training approaches also exploit model confidence to select reliable self-generated samples for further fine-tuning. For instance, STaR (Zelikman et al., 2022) and Meta-CoT (Xiang et al., 2025) bootstrap reasoning ability from a few labeled examples and reuse confident generations to improve performance iteratively. However, these methods rely on the generator's own confidence (often measured by softmax probabilities or consistency across samples), which can be unstable and prone to overconfidence in large language models. In contrast, our framework decouples confidence estimation from generation by introducing a separately trained reasoning-correctness classifier. This external verifier provides a more objective confidence measure, enabling more stable and transferable pseudo-label selection under minimal supervision.

2.3. Confidence Estimation and Sample Selection

Pseudo-labeling inherently risks introducing noisy or incorrect samples, which can degrade performance. To mitigate this, previous work has employed uncertainty-based filtering. Confidence-based pseudo-labeling (Arazo et al., 2020) and uncertainty estimation frameworks (Kendall and Gal, 2017; Alturki and Alsulami, 2025) often utilize entropy as a measure of reliability. Especially, Saporta et al. (2020) introduced entropy-guided self-supervised learning for semantic segmentation, using entropy as a confidence measure for pseudo-label selection. They demonstrate that entropy-based filtering produces more reliable pseudo labels than maximum-probability thresholding. Inspired by these studies, our framework applies an entropy-based confidence threshold to the classifier's predictions, enabling the model to dynamically select high-confidence reasoning samples while discarding unreliable ones.

2.4. Integration with Self-Verification Frameworks

Recent advances in self-verification and self-improvement of LLMs have introduced iterative reasoning refinement loops. Methods such as Self-Refine (Madaan et al., 2023), ReGenesis (Peng et al., 2025), S^2R (Ma et al., 2025), and Confident Reasoning (Jang et al., 2025) allow LLMs to generate, critique, and revise their own reasoning. However, most of these approaches let the same model act as both generator and verifier (*LLM-as-a-judge*) or rely on implicit confidence signals. In contrast, our work explicitly trains a separate reasoning-

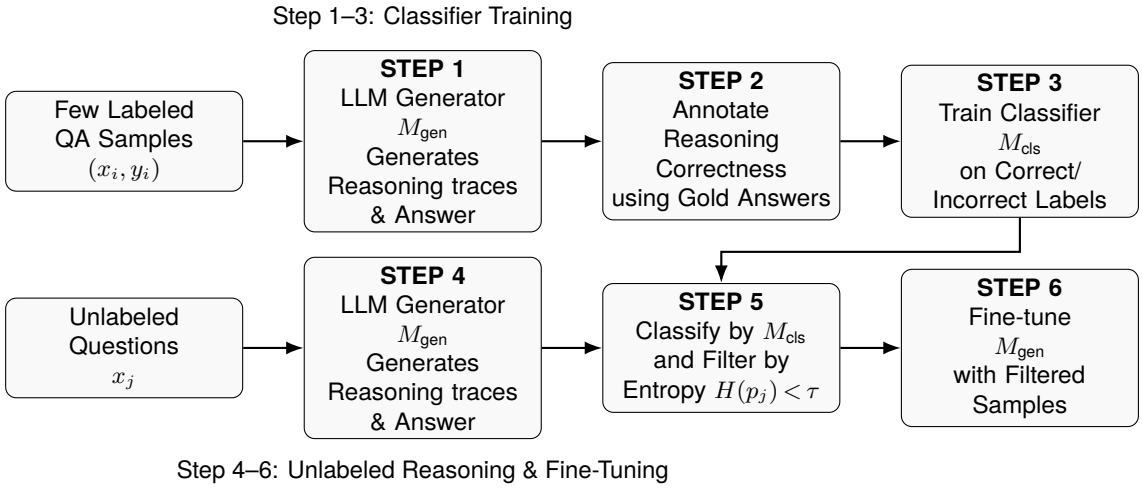


Figure 1: Overview of the proposed semi-supervised reasoning framework. **Top (Step 1–3):** From a few labeled QA samples, M_{gen} produces reasoning and answers; reasoning correctness is annotated using gold answers, and a reasoning-correctness classifier M_{cls} is trained. **Bottom (Step 4–6):** For unlabeled questions, M_{gen} generates reasoning; M_{cls} classifies them and low-entropy (high-confidence) samples are retained to fine-tune M_{gen}.

correctness classifier, which can be seamlessly incorporated into these frameworks as a transferable, lightweight discriminator.

This design is not in conflict with prior methods; rather, it complements them. By inserting an explicit reasoning-verification module into self-improvement loops, one can stabilize pseudo-label generation, suppress noise propagation, and enable semi-supervised reasoning learning even when explicit answer annotations are unavailable. Therefore, our framework provides a modular component that can further strengthen reinforcement-based reasoning optimization (e.g., using GRPO or PPO) and collaborative verification (Liang et al., 2024), bridging static pseudo-labeling and dynamic self-reinforcement.

3. Method

3.1. Overview and Intuition

We aim to improve LLM reasoning from minimal supervision by *learning to judge reasoning* and then *using that judgment to scale training data*. The core intuition is twofold: (i) for LLMs, **binary verification** ("is this reasoning correct?") is typically easier than **free-form generation** of correct reasoning; thus a small labeled set can train a useful verifier,[1] and (ii) **distributional uncertainty** (entropy) provides a more robust confidence signal than raw token probabilities, which are often overconfident and poorly calibrated (Guo et al., 2017; Kendall and Gal, 2017; Desai and Durrett, 2020; Kadavath et al., 2022).

The overview is shown in Figure 1. Operationally, we (1) generate reasoning on a few labeled QA pairs, (2) derive correctness labels from gold answers, (3) train an LLM-based correctness classifier, (4) generate reasoning on unlabeled problems, (5) filter by the classifier's entropy, and (6) fine-tune the generator on the selected pseudo labels.

An intuitive analogy is adversarial learning: in practice, a *discriminator* (binary judge) often learns faster and provides a useful training signal to a harder *generator*. This intuition is also aligned with the easy-to-hard generalization paradigm of Sun et al. (2024), which shows that an evaluation model trained on relatively simple, human-supervised tasks can be leveraged to progressively improve performance on more difficult tasks. We adopt the same spirit here—first learn a lightweight judge for reasoning, then use its confidence to curate pseudo labels—while deferring broader theoretical evidence to Section 3.4.

3.2. Problem Setup and Notation

Let $\mathcal{D}_{\text{lab}} = \{(x_i, y_i)\}_{i=1}^{N_\ell}$ be a small labeled set (problem, gold answer) and $\mathcal{D}_{\text{unlabeled}} = \{x_j\}_{j=1}^{N_u}$ a large unlabeled set. We use a generator M_{gen} and a verifier (classifier) M_{cls}. A *reasoning trace* is denoted r, and a generated answer $\hat{y}$.

3.3. Training Steps

Step 1: Reasoning Generation on Labeled Data. For each labeled pair $(x_i, y_i) \in \mathcal{D}_{\text{lab}}$, the genera-

[1] From cognitive psychology, *recognition/verification* is generally simpler and more reliable than *free recall/production* (Tenenbaum et al., 2011); our use of a binary verifier follows this asymmetry.

tor produces a reasoning trace and a candidate answer:

$$r_i,\ \hat{y}_i \ = \ M_{\text{gen}}(x_i).$$

This converts scarce labeled QA pairs into $(x_i, r_i, \hat{y}_i)$ triples that expose intermediate reasoning behavior.

Step 2: Correctness Annotation from Gold Answers. We derive a binary label $z_i \in \{0, 1\}$ indicating whether the produced reasoning/answer is correct given gold y_i, yielding a small verifier-training set:

$$\mathcal{D}_{\text{reason}} \ = \ \big\{(x_i,\, r_i,\, \hat{y}_i,\, z_i)\big\}_{i=1}^{N_\ell}.$$

Turning generation into binary feedback provides a simpler signal that can be learned from few examples.

Step 3: Training a Reasoning-Correctness Classifier. We fine-tune an LLM-based classifier M_{cls} to output `correct`/`incorrect` given $(x, r, \hat{y})$:

$$p(z{=}1 \mid x, r, \hat{y}) \ = \ M_{\text{cls}}\big(z{=}\texttt{correct} \mid x, r, \hat{y}\big).$$

The objective is binary cross-entropy over $\mathcal{D}_{\text{reason}}$:

$$\mathcal{L}_{\text{cls}} = -\sum_{\substack{(x_i, r_i, \hat{y}_i, z_i) \\ \in \mathcal{D}_{\text{reason}}}} \Big[z_i \log p_i + (1{-}z_i)\, \log\big(1{-}p_i\big) \Big],$$

where $p_i \ = \ M_{\text{cls}}\big(z{=}\texttt{correct} \mid x_i, r_i, \hat{y}_i\big)$. This discriminative training calibrates relative confidence more reliably than relying on raw generation scores (Desai and Durrett, 2020; Kadavath et al., 2022).

Step 4: Reasoning Generation on Unlabeled Problems. For each $x_j \in \mathcal{D}_{\text{unlabeled}}$, the generator produces:

$$r_j,\ \hat{y}_j \ = \ M_{\text{gen}}(x_j).$$

The triples $(x_j, r_j, \hat{y}_j)$ are passed to the verifier in Step 5.

Step 5: Confidence-Based Filtering via Entropy. Given $(x_j, r_j, \hat{y}_j)$, the classifier outputs the probability that the reasoning is correct:

$$p_j \ = \ M_{\text{cls}}\big(z{=}\texttt{correct} \mid x_j, r_j, \hat{y}_j\big).$$

Because token probabilities are often overconfident (Guo et al., 2017), we use the verifier's entropy as a reliability score:

$$H(p_j) \ = \ -\Big[p_j \log p_j \ + \ (1{-}p_j) \log(1{-}p_j)\Big].$$

We select low-entropy (high-confidence) samples:

$$\mathcal{D}_{\text{pseudo}} \ = \ \big\{(x_j, r_j, \hat{y}_j) \ \big|\ H(p_j) < \tau\big\}.$$

Entropy captures distributional uncertainty and yields higher-precision pseudo labels in practice (Figure 2; Sec. 4.5).

Step 6: Fine-Tuning the Generator with Selected Pseudo Labels. We fine-tune M_{gen} on $\mathcal{D}_{\text{lab}} \cup \mathcal{D}_{\text{pseudo}}$ via standard supervised fine-tuning (SFT), treating selected reasoning traces as training signals. In summary, the method converts a few labeled QA pairs into a reliable reasoning verifier, uses entropy-calibrated judgments to harvest high-precision pseudo reasoning from large unlabeled pools, and finally fine-tunes the generator on the selected set—realizing scalable, semi-supervised improvement of reasoning with minimal supervision.

3.4. Additional Theoretical Support

Our design is further supported by prior findings that emphasize the relative tractability of *verification/selection* over *generation*. Self-Consistency improves performance by *evaluating and selecting* among multiple reasoning traces rather than requiring a single perfect derivation (Wang et al., 2023). Self-Refine–style frameworks operationalize iterative *critique and revision*, making verification a simpler subproblem that guides generation (Madaan et al., 2023). In parallel, calibration studies indicate that large models provide *useful relative confidence signals* even when textual generations are imperfect (Guo et al., 2017; Desai and Durrett, 2020; Kadavath et al., 2022). Together, these results support our approach of (i) training a binary verifier from few labels and (ii) using its entropy-based confidence to select high-precision pseudo reasoning for scalable semi-supervised learning.

4. Experiments

4.1. Experimental Setting and Motivation

Our goal is to empirically verify whether the proposed framework can improve reasoning performance under extremely limited annotation conditions. Although our experiments use relatively small models and datasets, they are designed to rigorously test the *principle* of the method rather than to compete on scale. Each experiment compares: (1) SFT with a small number of labeled samples, (2) our proposed semi-supervised framework with reasoning verification and entropy filtering, and (3) SFT with a larger number of labeled samples as an upper bound.

Setting	Labeled	Unlabeled	Filtered	Total	Accuracy
Zero-shot	–	-	-	-	25.2
SFT (200 labels)	200	0	0	200	26.6
Proposed (200 labels + 100k unlabeled)	200	100,000	2,911	3,111	**30.1**
SFT (3,000 labels)	3,000	0	0	3,000	30.6

Table 1: Results on Verifiable Math Problems (Orca-Math subset). **Labeled**, **Unlabeled**, **Filtered**, and **Total** indicate the number of samples. **Filtered** represents the number of samples remaining after filtering the **Unlabeled** data using M_{cls}. **Total** is the final sample number for training (**Labeled** + **Filtered**).

4.2. Dataset 1: Verifiable Math Problems (Orca-Math Subset)

Task Overview. We use the *verifiable-math-problems* dataset on HuggingFace, specifically the *Orca-Math* subset, which contains mathematical reasoning problems paired with intermediate reasoning and final answers. We evaluate models using answer accuracy.

We adopt *Orca-Math* as a representative mathematical reasoning benchmark to examine whether our semi-supervised framework can improve reasoning ability on general, language-based problem solving tasks that are widely used for evaluating large language models.

Furthermore, several popular mathematical datasets, including *GSM8K* and *MATH*, have been observed to show little or no improvement from SFT, and previous works have raised concerns about possible data leakage from pre-training corpora. In contrast, *Orca-Math* exhibits consistent performance gains under SFT, suggesting that the influence of data leakage is relatively minor. This makes it an appropriate and reliable testbed for evaluating the effectiveness of semi-supervised reasoning learning under limited supervision.

It is also worth noting that publicly available mathematical reasoning datasets that are both general-purpose and largely unaffected by pre-training leakage remain scarce. In this context, *Orca-Math* offers a balanced choice between experimental clarity and general applicability.

Setup. We randomly split 500 samples for testing and 500 for validation. We set the number of labeled QA pairs to 200 to balance *practical* annotation budgets in expert domains and *statistical* adequacy for a binary verifier. In practice, specialist supervision is costly and typically available only in the low hundreds—e.g., biomedical QA benchmarks such as *PubMedQA* (Jin et al., 2019) and the *BioASQ* challenge (Tsatsaronis et al., 2015) rely on expert involvement and limited high-quality labels; similarly, legal NLP benchmarks aggregate expert-curated tasks under constrained supervision (e.g., *LexGLUE* (Chalkidis et al., 2022)). Alignment-style supervision for general LMs is also carefully budgeted (Ouyang et al., 2022). Taken together, few-hundred–scale seeds are a realistic starting point when labels must be expert-provided or statistically robust.[2]

The base model is Qwen2.5-0.5B-Instruct (Qwen, 2024), a widely used open-source instruction-tuned LLM. In this work, to investigate the self-improvement capabilities of LLMs, the same base model is employed for both the generator M_{gen} and the classifier M_{cls}. We intentionally adopt the smallest available configuration, as scenarios involving limited labeled data often coincide with low computational budgets in practical applications. Evaluating our method under such a constrained setting allows us to assess whether the proposed semi-supervised framework remains effective even when both annotation and computational resources are scarce.

We use the trl library (Werra et al., 2020) for training. For the reasoning-correctness classifier M_{cls}, early stopping is applied based on the classification accuracy on the validation set. For the final reasoning generator M_{gen}, early stopping is determined by the answer accuracy on the validation set. The batch size is set to 64, the learning rate to 1×10^{-6}, and the warm-up ratio to 0.05. These hyperparameters are empirically determined.

Results. The result is shown in Table 1. With only 200 labeled samples used to train the reasoning classifier, the model filtered reasoning traces from 100,000 unlabeled problems and selected the top 10% most confident samples (based on entropy). This entropy threshold is determined by the observation described in Section 4.5. This achieved 30.1% accuracy—comparable to SFT using 3,000 labeled samples. This result empirically supports our hypothesis that LLMs can more easily learn to *judge* reasoning correctness than to generate reasoning itself. Note that the same 200 labeled samples used for training the classifier were also reused in the final model fine-tuning.

[2]For a Bernoulli accuracy estimate of a binary verifier, a normal-approximation suggests $n \approx z^2_{1-\alpha/2}\, p(1-p)/\varepsilon^2$. Under the conservative worst case $p{=}0.5$ with $z_{0.975}{=}1.96$ and target margin $\varepsilon{=}0.07$, one obtains $n{\approx}196$ (about 200), providing a stable validation/early-stopping signal without excessive annotation burden.

Setting	Labeled	Unlabeled	Filtered	Total	Accuracy
Zero-shot	–	-	-	-	52.2
Khan et al. (SFT, 200 labels)	200	0	0	200	51.8
Proposed (200 labels + 91k unlabeled)	200	91,300	3,551	3,751	**54.1**
Khan et al. (SFT, 2,000 labels)	2,000	0	0	2,000	54.5

Table 2: Results on GQA with Visual Programming.

4.3. Dataset 2: GQA with Visual Programming

Task Overview. To test the generality of our approach in a multi-modal setting, we follow prior work on *Visual Programming* (Gupta et al., 2023; Khan et al., 2024). GQA (Hudson and Manning, 2019) is a challenging Visual Question Answering (VQA) benchmark requiring compositional reasoning over visual scenes. This setup represents a case where the model must learn a new form of reasoning that is not typically covered by general-purpose pre-training corpora, thereby testing whether the proposed semi-supervised scheme can extend LLM reasoning to previously unseen modalities and tasks. Example of GQA with Visual Programming is shown as following.

Question: *Are there both ties and glasses in the picture?*

Program (pseudo-code):

```
BOX0=Loc(image=IMAGE,object='tie')
ANSWER0=Count(box=BOX0)
BOX1=Loc(image=IMAGE,object='glass')
ANSWER1=Count(box=BOX1)
ANSWER2=Eval("yes" if (ANSWER0) > 0
and (ANSWER1) > 0 else "no")
RESULT=ANSWER2
Prediction: no
```

This example illustrates the visual program's compositional structure: object localization (`Loc`), quantitative queries (`Count`), and symbolic combination via logical evaluation (`Eval`). Our setting trains an LLM to generate such programs, while a separate vision-language executor runs them to obtain the final answer.

Because the dataset does not include intermediate reasoning annotations, direct SFT with reasoning supervision is not feasible. Instead, we build on the self-training strategy of Khan et al. (2024), where pseudo-programs (object detection → attribute extraction → relation reasoning → answer) are generated in a zero-shot manner, executed, and only those that yield correct final answers are retained for fine-tuning. Importantly, the final GQA question-answering task is solved by a vision-language model separate from the program-generating LLM, allowing us to isolate the quality of generated reasoning programs from the perception and answering capabilities of the underlying visual model.

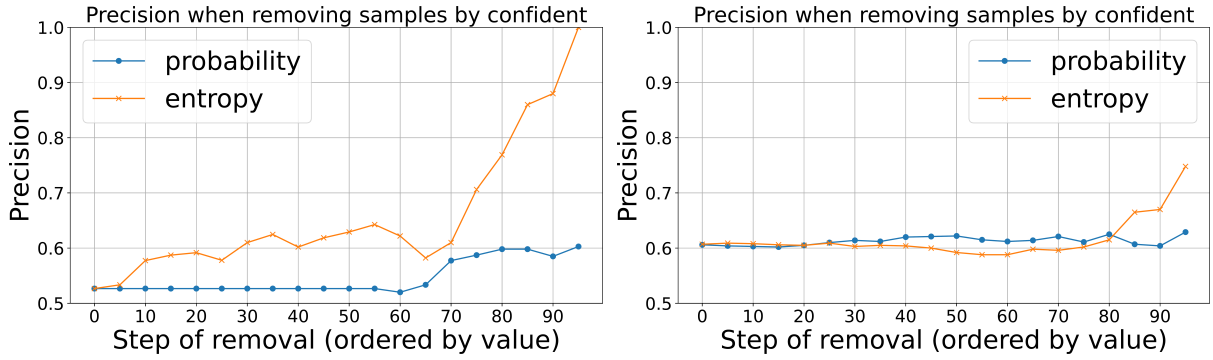
Setup. We adopt the same data split as prior work (Khan et al., 2024), using 1,912 samples for test and 891 for validation. The number of labeled QA pairs is set to 200 as well in Section 4.2.

The base model is Llama3-8B-Instruct (Grattafiori et al., 2024) for both the generator M_{gen} and the classifier M_{cls}. Similar to Qwen2.5-0.5B used in the previous experiment, this model is a widely adopted open-source instruction-tuned LLM. We select it because its scale is representative of commonly used mid-sized models, and evaluating our method on such a model allows us to assess the generality of the proposed framework across different architectures and parameter scales.

Zero-shot prompting uses 27 in-context examples as in Khan et al. (2024). For our framework, we trained a reasoning-classification model on 200 labeled samples and applied it to 91,300 unlabeled questions, selecting the top 10% most confident samples by entropy. This entropy threshold is determined by the observation described in Section 4.5 as well.

We use the Hugging Face Trainer for training. For the reasoning-correctness classifier M_{cls}, early stopping is applied based on the classification accuracy on the validation set. For the reasoning generator M_{gen} in the GQA setting, the final task accuracy depends on a separate vision-language execution engine; therefore, early stopping is determined by the validation loss of the generator itself rather than downstream answer accuracy. The batch size is set to 1, the learning rate to 1×10^{-3}. Gradient accumulation steps is set to 4. These hyperparameters are empirically determined. The rest of the hyperparameters are the default values for each library. For executing the VQA task, we employ BLIP (Li et al., 2022) as the engine, and OwlVIT (Minderer et al., 2022) is used for object detection.

Results. Table 2 summarizes the results of the baseline and proposed methods. Our method improves over both zero-shot reasoning and the self-training baseline with only 200 labeled samples, approaching the performance of SFT using 2,000 labeled samples. This demonstrates that the proposed reasoning-verification framework can effectively identify and exploit reliable intermediate reasoning traces even when the dataset lacks



(a) Orca-Math (Qwen2.5-0.5B-Instruct) (b) GQA (Llama3-8B-Instruct)

Figure 2: Precision of pseudo-labeled reasoning samples under different confidence metrics. Each plot shows precision as a function of the exclusion ratio of low-confidence samples (x-axis: percentage of samples removed from least to most confident, y-axis: precision). Filtering by the final-token probability (blue) yields no or very small improvement, whereas entropy-based filtering (orange) produces a sharp precision rise around the 90% exclusion point, where only the top 10% of samples remain.

explicit annotations. Importantly, this shows that the classifier-based filtering approach is complementary to visual reasoning pipelines such as Visual Programming, highlighting the potential of our method to generalize beyond textual reasoning tasks.

4.4. Overall Discussion

Although the experimental scale is modest, the results consistently demonstrate the principle behind our approach: (i) Even with a very small number of answer annotations, reasoning quality can be improved through classifier-based filtering. (ii) The method achieves comparable gains to increasing labeled data by an order of magnitude. (iii)Intermediate reasoning can be treated as a renewable data resource, supporting efficient semi-supervised learning. Rather than emphasizing absolute performance, we view these results as a *proof-of-concept* confirming that learning to judge reasoning correctness is a feasible and efficient path toward scalable reasoning improvement, especially in domains where labeling is costly.

4.5. Effect of the Entropy-Based Confidence Filter

A key finding across both datasets is that the entropy-based confidence filtering mechanism substantially improves the quality of pseudo-labeled reasoning samples. To visualize this effect, we analyze how the precision of selected samples changes as the confidence threshold becomes stricter. Figure 2 shows validation precision when filtering by either the final-token probability (blue) or the entropy of the classifier output (orange), as low-confidence samples are progressively excluded (i.e., retaining only the top-k% most confident ones).

As illustrated, precision remains nearly flat when filtering by raw probability, indicating that the classifier's predicted probabilities are not well calibrated and tend to overestimate confidence. In contrast, entropy-based filtering yields a clear and consistent improvement: precision sharply increases when only the top 10% of lowest-entropy samples are retained. This behavior is observed in both (a) Orca-Math with Qwen2.5-0.5B and (b) GQA with Llama3-8B, demonstrating that entropy provides a more stable confidence signal for pseudo-label selection across architectures and modalities. This observation is consistent with the previous work in a different task domain (Saporta et al., 2020).

Discussion. This analysis explains why entropy-based filtering was adopted in our main experiments. Unlike raw probability thresholds, entropy better reflects distributional uncertainty, leading to higher-precision pseudo reasoning samples and more stable semi-supervised training behavior. The precision curves also provide a principled basis for selecting the 10% high-confidence cutoff used throughout the paper.

4.6. Ablation Study

We further analyze the contribution of each component of our framework. Specifically, we investigate: (1) whether the reasoning-classifier itself is necessary, (2) how the number of unlabeled samples affects performance, and (3) the impact of the entropy-based confidence filter.

Setting	labeled	Unlabeled	Filtered	Total	Accuracy
Zero-shot	–	-	-	-	25.2
w/o Classifier (100k)	200	100,000	100,000	100,200	25.6
Classifier (10k)	200	10,000	3,440	3,640	27.2
Classifier (10k, top 10%)	200	10,000	285	305	28.2
Classifier (100k)	200	100,000	34,731	34,931	27.4
Classifier (100k, top 10%)	200	100,000	2,911	3,111	**30.1**

Table 3: Ablation results on Verifiable Math Problems (Orca-Math subset). The reasoning classifier and entropy-based filtering consistently improve accuracy.

Setting	labeled	Unlabeled	Filtered	Total	Accuracy
Zero-shot	–	-	-	-	52.2
w/o Classifier (91k)	200	91,130	91,130	91,330	52.0
Classifier (9k)	200	9,113	3,923	4,123	53.4
Classifier (9k, top 10%)	200	9,113	338	538	52.2
Classifier (91k)	200	91,130	42,152	42,352	53.6
Classifier (91k, top 10%)	200	91,130	3,551	3,751	**54.1**

Table 4: Ablation results on GQA with Visual Programming. Entropy filtering enhances data efficiency, particularly when scaling to large unlabeled corpora.

4.6.1. Effect of the Reasoning Classifier and Entropy Filtering

Verifiable Math Problems (Orca-Math). Table 3 summarizes the results under different unlabeled data sizes and filtering conditions. Without the reasoning classifier, simply augmenting training data with model-generated reasoning does not lead to improvement. Introducing the classifier already provides clear gains, and adding entropy-based filtering further boosts accuracy, especially at larger unlabeled scales.

Discussion. The results clearly indicate that naive self-training without a reasoning-quality classifier (row 2) fails to improve upon the zero-shot baseline. In contrast, classifier-guided pseudo-labeling improves performance steadily with increasing unlabeled data. Moreover, entropy-based filtering raises the performance ceiling: it prevents degradation from noisy pseudo reasoning and enables higher precision in reasoning supervision. Even when the overall number of samples is large, without confidence filtering the improvement quickly saturates, while the entropy filter continues to yield marginal gains.

GQA with Visual Programming. A similar trend is observed in the visual reasoning task (Table 4). Because GQA does not contain explicit intermediate reasoning annotations, the classifier acts as a crucial proxy for reasoning verification. We again find that entropy filtering stabilizes training and allows the model to selectively focus on reliable pseudo reasoning.

Discussion. For GQA, increasing unlabeled data improves performance only up to a certain saturation point. Entropy filtering boosts the accuracy further by removing uncertain pseudo labels, suggesting that the optimal trade-off between confidence threshold and data volume depends on the task complexity and reasoning diversity. Nevertheless, since question-only data are inexpensive to collect, gradually expanding unlabeled samples with entropy-based filtering is a practical way to scale reasoning learning in real-world settings.

4.6.2. Summary of Findings

Across both tasks, the following trends hold consistently: (1) Without the reasoning classifier, pseudo reasoning introduces noise and yields minimal benefit. (2) The reasoning classifier alone significantly improves data efficiency. (3) Entropy filtering raises the upper performance bound and mitigates saturation. (4) The trade-off between sample count and confidence threshold is task-dependent, but the general principle of selective pseudo-labeling remains robust.

These ablations reinforce that our method's strength lies not in brute-force data expansion, but in leveraging LLMs' inherent ability to *discriminate* correct reasoning from incorrect reasoning, enabling efficient semi-supervised scaling with minimal human annotation.

5. Conclusion

This paper presents a semi-supervised framework that improves LLM reasoning with minimal supervision by *learning to judge reasoning*: a correctness classifier, trained on few labeled examples,

filters reliable reasoning traces from large unlabeled pools. On textual (Orca-Math) and multimodal (GQA) tasks, the method matches the performance of using 10–15$\times$ more labeled data, indicating that *judging* is a data-efficient proxy for *generating* reasoning.

Our ablation studies further revealed that: (1) without the classifier, naive pseudo-labeling brings little gain; (2) entropy-based filtering prevents saturation and raises the upper performance bound; and (3) the optimal balance between confidence and quantity depends on task complexity. Together, these results highlight that selective self-training—guided by confidence and reasoning discrimination— is more effective than brute-force data expansion.

Future Outlook. We envision autonomous reasoning learning systems capable of self-improvement through problem generation, reasoning, and verification, eliminating human annotation. This closed-loop approach, integrating reasoning verification with automatic problem generation, offers a scalable path for self-evolving LLMs to achieve lifelong reasoning acquisition.

Ethics Statement

We use the public datasets Verifiable Math Problems (Orca-Math subset) and GQA strictly under their licenses for research purposes. Both datasets are provided without personally identifiable information; problem statements, (when applicable) intermediate reasoning, and final answers are already included in the datasets, and we did not conduct any new human annotation. Because data leakage into pre-training corpora is a known concern for mathematical benchmarks, we primarily adopt Orca-Math, where SFT yields observable gains, to comparatively reduce leakage effects. While our method can expand pseudo reasoning from few labels, it can also propagate erroneous or biased reasoning. We mitigate this risk via an explicit classifier and entropy-based confidence filtering. However, for applications in high-stakes domains (e.g., medical or legal), human oversight and validation are strongly recommended.

Limitations

While our study demonstrates the potential of reasoning verification for semi-supervised learning with large language models, several limitations should be acknowledged.

Experimental Scale. The experiments were conducted on relatively small models (*Qwen2.5-0.5B-Instruct* and *Llama3-8B-Instruct*) and limited datasets (Orca-Math and GQA subsets). Our goal was to validate the core principle rather than absolute performance. Nevertheless, larger-scale evaluations are necessary to confirm the generality and robustness of the proposed framework across domains and architectures.

Entropy-Based Thresholding. The confidence estimation component in this work relies solely on Shannon entropy as a simple uncertainty measure. Although effective, this criterion does not account for model calibration, multi-modal uncertainty, or hierarchical dependencies among reasoning steps. Future work could explore adaptive thresholding, Bayesian confidence estimation, or energy-based filtering to better balance coverage and precision in pseudo reasoning selection.

Learning Framework. The proposed method is implemented in a supervised fine-tuning (SFT) setting, without incorporating reinforcement signals. While this choice ensures stability and reproducibility, the framework could be extended to reinforcement learning formulations such as GRPO or PPO-based reasoning optimization, where the reasoning classifier serves as a reward model. Integrating reasoning verification into reinforcement loops may allow continuous self-improvement beyond static pseudo labeling.

Scope of Reasoning. Finally, our approach has been tested only on tasks where intermediate reasoning is expressed in textual or structured form. Extending this paradigm to free-form or multi-modal reasoning (e.g., diagrammatic or procedural reasoning) requires further investigation into how classifiers interpret and evaluate non-textual reasoning traces.

Despite these limitations, we believe the simplicity and extensibility of the proposed framework make it a promising foundation for future research in scalable, self-supervised reasoning systems.

6. Bibliographical References

Badraddin Alturki and Abdulaziz A. Alsulami. 2025. Semi-supervised learning with entropy filtering for intrusion detection in asymmetrical iot systems. *Symmetry*, 17(6):973.

Eduardo Arazo, Diego Ortego, Paul Albert, Noel E. O'Connor, and Kevin McGuinness. 2020. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

Ilias Chalkidis, Abhik Jana, Dirk Hartung, and Ion Androutsopoulos. 2022. Lexglue: A benchmark dataset for legal language understanding in english. In *NeurIPS Datasets and Benchmarks Track*.

Shrey Desai and Greg Durrett. 2020. Calibration of pre-trained transformers. In *Proceedings of the 2020 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, and et al. 2024. The llama 3 herd of models.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*.

Tanmay Gupta, Akilesh Gotmare, Kevin Shih, Anand Mishra, Rama Chellappa, and Derek Hoiem. 2023. Visual programming: Compositional visual reasoning without training. In *CVPR*.

Drew A. Hudson and Christopher D. Manning. 2019. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Hyosoon Jang, Yunhui Jang, Sungjae Lee, Jungseul Ok, and Sungsoo Ahn. 2025. Self-training large language models with confident reasoning. *arXiv preprint arXiv:2505.17454*.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William W. Cohen, and Xinghua Lu. 2019. Pubmedqa: A dataset for biomedical research question answering. In *EMNLP*.

Saurav Kadavath, Andy Lin, Rylan Schaeffer, Jacob Hilton, James McLean, Laria Wu, Ilya Kostrikov, Dario Amodei, Catherine Olsson, Paul Christiano, et al. 2022. Language models (mostly) know what they know. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Alex Kendall and Yarin Gal. 2017. What uncertainties do we need in bayesian deep learning for computer vision? In *Advances in Neural Information Processing Systems (NeurIPS)*.

Zeeshan Khan, Kevin Lin, Jae Sung Park, Peter Anderson, Yixin Nie, Ankit Goyal, Zhenhai Zhu, Mengye Ren, and Mohit Bansal. 2024. Self-training large language models for improved visual program synthesis with vision-language feedback. In *CVPR*.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation.

Zhenwen Liang, Ye Liu, Tong Niu, Xiangliang Zhang, Yingbo Zhou, and Semih Yavuz. 2024. Improving llm reasoning through scaling inference computation with collaborative verification.

Ruotian Ma, Peisong Wang, Cheng Liu, Xingyan Liu, Jiaqi Chen, Bang Zhang, Xin Zhou, Nan Du, and Jia Li. 2025. S^2r: Teaching llms to self-verify and self-correct via reinforcement learning. *arXiv preprint arXiv:2502.12853*.

Aman Madaan, Shuyan Zhou, Uri Alon, Yiming Yang, and Graham Neubig. 2023. Self-refine: Iterative refinement with self-feedback. *arXiv preprint arXiv:2303.17651*.

Matthias Minderer, Alexey Gritsenko, Austin Stone, Maxim Neumann, Dirk Weissenborn, Alexey Dosovitskiy, Aravindh Mahendran, Anurag Arnab, Mostafa Dehghani, Zhuoran Shen, Xiao Wang, Xiaohua Zhai, Thomas Kipf, and Neil Houlsby. 2022. Simple open-vocabulary object detection with vision transformers.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishra, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. In *NeurIPS*.

Xiangyu Peng, Congying Xia, Xinyi Yang, Caiming Xiong, Chien-Sheng Wu, and Chen Xing. 2025. Regenesis: Llms can grow into reasoning generalists via self-improvement. *arXiv preprint arXiv:2410.02108*.

Team Qwen. 2024. Qwen2.5: A party of foundation models.

Antoine Saporta, Tuan-Hung Vu, Matthieu Cord, and Patrick Pérez. 2020. Esl: Entropy-guided self-supervised learning for domain adaptation in semantic segmentation. *arXiv preprint arXiv:2006.08658*.

Noah Shinn, Peter West, and Yejin Choi. 2023. Reflexion: Language agents with verbal reinforcement learning. *arXiv preprint arXiv:2303.11366*.

Zhiqing Sun, Longhui Yu, Yikang Shen, Weiyang Liu, Yiming Yang, Sean Welleck, and Chuang Gan. 2024. Easy-to-hard generalization: Scalable alignment beyond human supervision. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Joshua B. Tenenbaum, Charles Kemp, Thomas L. Griffiths, and Noah D. Goodman. 2011. How to grow a mind: Statistics, structure, and abstraction. *Science*, 331(6022):1279–1285.

George Tsatsaronis, Georgios Balikas, Prodromos Malakasiotis, Ioannis Partalas, Max Zschunke, Michael R. Alvers, Dirk Weissenborn, Anastasia Krithara, Sergios Petridis, Dimitris Polychronopoulos, et al. 2015. An overview of the bioasq large-scale biomedical semantic indexing and question answering competition. *BMC Bioinformatics*, 16(1):138.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed Chi, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations (ICLR)*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Leandro Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouédec. 2020. Trl: Transformer reinforcement learning. https://github.com/huggingface/trl.

Violet Xiang, Charlie Snell, Kanishk Gandhi, Alon Albalak, Anikait Singh, Chase Blagden, Duy Phung, Rafael Rafailov, Nathan Lile, Dakota Mahan, Louis Castricato, Jan-Philipp Franken, Nick Haber, and Chelsea Finn. 2025. Towards system 2 reasoning in llms: Learning how to think with meta chain-of-thought. *arXiv preprint arXiv:2501.04682*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023a. Tree of thoughts: Deliberate problem solving with large language models. *arXiv preprint arXiv:2305.10601*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Andrea Madotto, Izhak Shafran, and Karthik Narasimhan. 2023b. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*.

Eric Zelikman, Jesse Mu, and Noah D. Goodman. 2022. Star: Bootstrapping reasoning with reasoning. *arXiv preprint arXiv:2203.14465*.