ReflectEvo: Improving Meta Introspection of Small LLMs by Learning Self-Reflection

Jiaqi Li[1], Xinyi Dong[2], Yang Liu[1], Zhizhuo Yang[2], Quansen Wang[1,2], Xiaobo Wang[1], Songchun Zhu[1,2], Zixia Jia[1,✉], Zilong Zheng[1,✉]
[1]State Key Laboratory of General Artificial Intellligence, BIGAI
[2]Peking University
{lijiaqi, jiazixia, zlzheng}@bigai.ai

Abstract

We present a novel pipeline **ReflectEvo** to demonstrate that small language models (SLMs) can enhance meta introspection through reflection learning. This process iteratively generates self-reflection for self-training, fostering a continuous and self-evolving process. Leveraging this pipeline, we construct **ReflectEvo-460k**, a large scale, comprehensive self-generated reflection dataset with broadened instructions and diverse multi-domain tasks. Building upon this dataset, we demonstrate the effectiveness of **reflection learning** to improve SLMs' reasoning abilities using SFT and DPO with remarkable performance, substantially boosting Llama-3 from 52.4% to 71.2% and Mistral from 44.4% to 71.1%. It validates that ReflectEvo can rival or even surpass the reasoning capability of the three prominent open-sourced models on BIG-bench without distillation from superior models or fine-grained human annotation. We further conduct a deeper analysis on the high quality of self-generated reflections and their impact on error localization and correction. Our work highlights the potential of continuously enhancing the reasoning performance of SLMs through iterative reflection learning in the long run.

1 Introduction

Self-reflection involves meditating on, examining, and evaluating one's behaviors, thoughts, motivations, and desires (Atkins and Murphy, 1993; Von Wright, 1992; Denton, 2011). Typically, it inspects the reasoning process leading to the current solution, identifies errors in each step, generates critiques on the causes of the failure, and offers advice for refining the solution to improve the problem-solving performance of Large Language Models (LLMs) (Welleck et al., 2022; Ferraz et al., 2024; Li et al., 2024a; Tong Wu, 2024). Unlike the paradigm of learning directly from the reasoning process and final answer, we refer to it as the process of human-like *meta introspection*, which explicitly generates self-reflection, providing textual differentiation and gradients as clear critiques and guidance on what to learn and how to improve based on the current state.

Recent research has demonstrated that LLMs can self-improve through their intrinsic capability of self-reflection (Huang et al., 2022; Renze and Guven, 2024; Guo et al., 2025; Wang et al., 2024b). However, conventional approaches rely closely on LLMs with large model sizes or supervision distilled from a superior model. In this study, we challenge whether *the self-reflection capability of SLMs can be learned effectively from reflection data.* However, it usually requires high-cost on fine-grained human annotation to acquire high-quality data for fine-tuning and is impractical to scale. Therefore, we are also curious whether *it is possible to effectively utilize both high- and low-quality self-generated data from weaker models for reflection learning.* With this in mind, we aim to investigate the effectiveness of reflection learning via self-training (Luong et al., 2024; Qu et al., 2024; Pang et al., 2023; Tang et al., 2024) and further validate that the improvement of self-reflection can further strengthen LLM's inherent reasoning capabilities across various methods and tasks with more interpretability and generalization. We believe that this paradigm can act as a plug-and-play enhancement for various reasoning methods, which emulates human learning through a slower and deeper thought process that iteratively and ultimately derives self-evolution (Li et al., 2023; He et al., 2024; Li et al., 2024b; Tang et al., 2024, 2023).

Therefore, in this paper, we propose a novel pipeline **ReflectEvo** (Sec. 2), to automatically generate self-reflection data and leverage self-training to enhance LLM's reflection capability. To the best of our knowledge, we are the first to demonstrate the potential of **meta introspection** of LLMs that are asked to explicitly generate reflection as an in-

Findings of the Association for Computational Linguistics: ACL 2025, pages 16948–16966
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

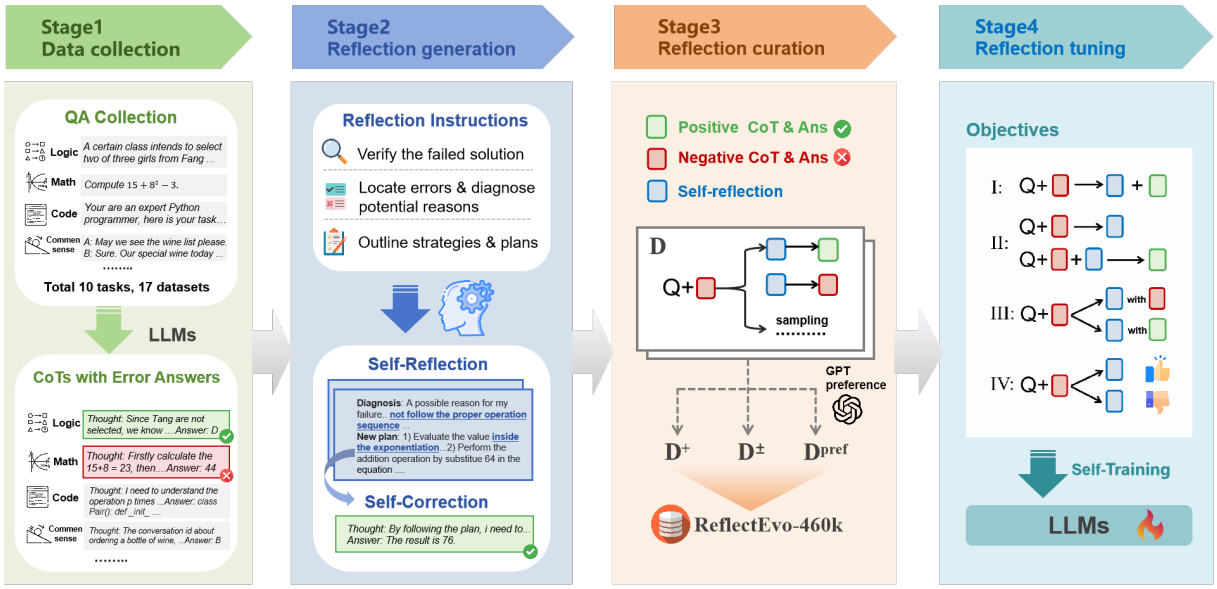


Figure 1: Overview pipeline of **ReflectEvo**.

termediate step-by-step process supervision rather than directly mapping an initial solution to a revised solution.

Building on this pipeline, we curate a large-scale, diverse, and unsupervised reflection learning dataset **ReflectEvo-460k** containing 460k reflection samples derived from 17 source datasets spanning 10 tasks and domains. We explore the diversity of reflection instructions and bootstrap multiple comparative reflections conditioned on the same question and initial solution. Based on the data, we develop **reflection learning** (Sec. 3) to further improve the self-reflection and self-correction capabilities of LLMs.

The evaluation results validate the effectiveness of reflection learning in boosting the reasoning of weak models. It shows significant improvements on Llama-3, exceeding the original base model by 10% on average tasks and outperforming its strongest counterpart with model size $\times 8$. We conduct a deeper analysis of the self-generated reflection data including various error types identified from the reflection and observe their gains on corrected answers.

In summary, our main contributions are:

- **Novel Pipeline for Self-Reflection Generation**: We propose ReflectEvo for automatic self-reflection generation and curation, which is the first to explore meta introspection of SLMs.
- **Large-Scale and Diverse Self-generated Reflection Dataset**: We curate a comprehensive reflection training set ReflectEvo-460K from multiple data sources and tasks including various reflection instructions and comparative samples.
- **Learning Reflection Via Self-training**: We develop four settings of reflection learning methods to effectively improve self-reflection and self-correction based on SFT and DPO, which significantly boost the reasoning abilities of SLMs as well as surpassing their stronger counterparts.

2 The ReflectEvo Generation Scheme

In this section, we introduce the end-to-end pipeline **ReflectEvo** for collecting self-generated reflections as the training data for Sec. 3, leveraging the inherent ability of SLMs (see Fig. 1).

2.1 Problem Definition and Prelinminary

Given a question q and its ground truth answer a^*, the answer of the LLM after reasoning is denoted as a followed by its corresponding verbal feedback f from the environment, where f represents the evaluation function that assesses whether an answer is correct or incorrect by comparing it to the reference answer a^*. The self-reflection r of an LLM explicitly locates and analyzes errors in a and makes further plans to mitigate the errors. Based on r and the context provided in the previous stage, the LLM is then asked to revise its original answer a to obtain $\hat{a}$ and solve q as correctly as possible.

2.2 Reflection Generation

Step 1: Collection of instruction pool To enhance the effectiveness and quality of the generated reflections r, we design instructions that target three key stages of reflection and correction, as

Feature	Logical Reasoning	Mathematics	Coding	Contextual QA	Context-Free QA	Reading Comprehension	Commonsense Reasoning	Social Reasoning	Causal Reasoning	Physics Reasoning	Total
# Reflection training samples	253,405	92,967	9,125	19,399	3,624	32,135	22,044	19,175	8,757	1,168	461,799
# Q&A-Reflection samples	164,746	106,434	7,520	17,448	2,940	20,760	10,012	11,404	3,212	468	344,944
% Correct after reflection	16.60	10.77	15.31	4.26	9.66	12.39	33.88	19.99	39.98	38.46	13.93
# Avg. question length	140	77	99	335	200	148	219	52	118	39	130
# Avg. answer length (turn 1)	131	267	187	91	118	82	158	110	118	112	189
# Avg. answer length (turn 2)	163	299	202	119	135	116	159	133	130	124	237
# Avg. reflection length	238	222	254	267	251	261	268	252	259	256	250

Table 1: Statistics of **ReflectEvo-460k**. The average length in the table is computed by tokens.

defined below: **i. Verify the failed solution.** It analyzes the initial solution by tracing and examining the reasoning process with or without step-by-step verification. **ii. Locate errors and diagnose potential reasons.** It points out errors in specific reasoning steps and identifies the causes (Zeng et al., 2024). We delicately design prompts to mitigate the most common error types (Li et al., 2024c), including mathematical (calculation & algorithm) errors, logic and reasoning errors (flawed rationale & internal inconsistency), instruction violation (context misinterpretation, incomplete or irrelevant response & format discrepancy), factual errors. They are explicitly specified in the instructions for error elimination and accurate fault localization. **iii. Outline strategies and plans for error correction and mitigation.** It provides strategies and guidance to address the error by proposing a high or low-level plan to mitigate similar issues in the future.

Step 2: Data generation Based on the instructions outlined in Step 1, we introduce two components for reflection generation: a **Generator** G (reasoning model) that generates the initial answer with its reasoning process and a **Reflector** R (reflection model) that improves the incorrect answer through self-reflection and self-correction.

Generator G Given a q, G is built upon a base LLM instructed to generate interleaved thoughts and an initial answer $G(a|q)$. It is implemented as described in ReAct (Yao et al., 2022), as the first step for self-reflection. We obtain the external environment feedback f by evaluating the correctness of a as a verifier. f is a binary signal "correct/incorrect" with limited information, which is usually the case in real scenarios, eliminating the need for enriched feedback from humans or more powerful models. If correct, a is directly used as the final answer. If incorrect, R is used to revise the solution iteratively. In this paper, we perform self-reflection once to maximize the efficiency of self-generated data; however, this approach can be extended to multiple iterations in future studies.

Reflector R We use exactly the same base LLM as G for R. The generation process for R is decomposed into **two phases: self-reflection and self-correction.** Self-reflection generates $R(r|q, a, f)$ to identify errors in the reasoning process and conduct a deeper analysis of the causes. Self-correction refines a as $R(\hat{a}|q, a, f, r)$. To enrich the self-training data, we sample k solution $\{r_j, \hat{a}_j\}_{j=1}^{k}$ for each $\{q, a, f\}$ conditioned on one specific prompt using reject sampling (Liu et al., 2023) to enrich the self-training data. We vary the prompts selected from the instruction pool to generate diverse self-reflection samples.

2.3 Reflection Curation

After the above-mentioned process, we obtain a reflection training set with $M(N * k * m)$ samples:

$$\mathcal{D} = \{q_i, a_i, f_i, (r_{i,j}, \hat{a}_{i,j})_{j=1}^{k*m}\}_{i=1}^{N}, \quad (1)$$

where N is the number of QA pairs in $\mathcal{D}$, m is the number of reflection instructions from pool, and k is the value of reject sampling. We aim to further curate the data for reflection learning as follows.

First, we filter r to include those followed by the correct $\hat{a}$, indicating that these reflections are of high quality for error correction, denoted as $\mathcal{D}^{+}$:

$$\mathcal{D}^{+} = \{(q_i, a_i, f_i, r_i, \hat{a}_i) \mid (\hat{a}_i = a^*)\}_{i=1}^{|\mathcal{D}|} \quad (2)$$

Subsequently, we leverage GPT-4o (Hurst et al., 2024) as a stronger teacher model to further select preferred reflection data from $\mathcal{D}^{+}$ to create pairwise data, denoted as $\mathcal{D}^{\text{pref}}$:

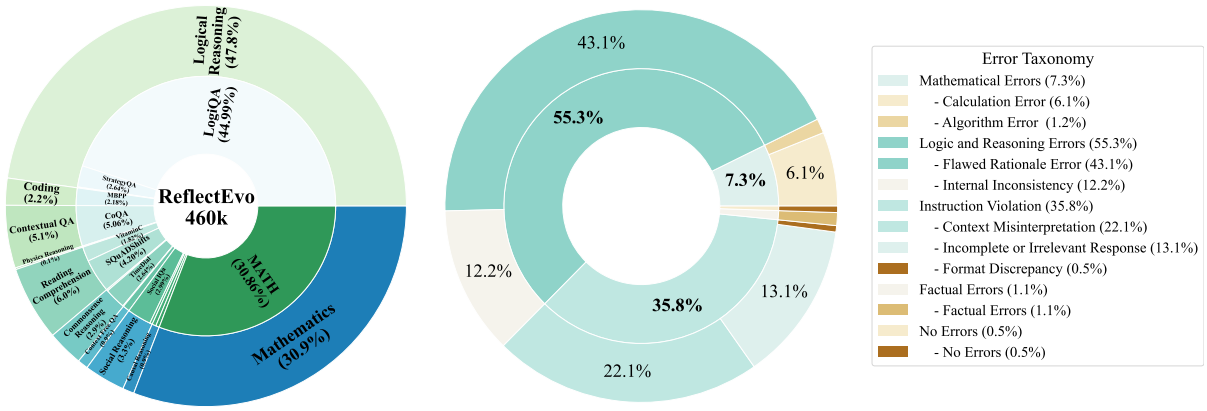
$$\mathcal{D}^{\text{pref}} = \{(q_i, a_i, f_i, [y_i^{\text{cho}}, y_i^{\text{rej}}]) \mid \exists\, y_i^{\text{cho}}, y_i^{\text{rej}}\}_{i=1}^{|\mathcal{D}^{+}|}, \quad (3)$$

where $y = (r, \hat{a})$ is the reflection and corresponding corrected answer. y^{cho} and y^{rej} are solutions randomly selected for each $\{q, a, f\}$ whose r is chosen and rejected, respectively, by GPT-4o.

To fully utilize low-quality reflection data followed by $\hat{a}_i$ that is still judged to be incorrect, we enrich the self-training data by incorporating both positive and negative samples as pairwise data for each $\{q, a, f\}$, denoted as $\mathcal{D}^{\pm}$.

$$\mathcal{D}^{\pm} = \{(q_i, a_i, f_i, [y_i^{+}, y_i^{-}]) \mid \exists\, y_i^{+}, y_i^{-}\}_{i=1}^{|\mathcal{D}|}, \quad (4)$$

where y^{+} and y^{-} are solutions whose $\hat{a}$ is evaluated as correct or incorrect by a^*.



(a) (b)

Figure 2: (a) Task-dataset hierarchy distribution of ReflectEvo-460k. (b) Error type distribution of corrected thoughts identified by reflection in the test sets.

Following the above-mentioned steps in Sec. 2, We create a reflection training dataset **ReflectEvo-460k** by curating examples of 17 carefully selected source subsets from LogiQA (Liu et al., 2020), MATH (Hendrycks et al., 2021), MBPP (Austin et al., 2021), and BIG-bench (bench authors, 2023), spanning diverse domains and categories. The Statistics of the dataset are shown in Tab. 1 and Fig. 2a. We use three commonly used SLMs including Llama-3-8B (Dubey et al., 2024), Mistral-7B (Jiang et al., 2023) and Gemma-2-9B (Team et al., 2024) for the entire process of data generation, training, and test. Implementation details and instructions are provided in Appendices B.1 and C.

3 Reflection Learning on Self-Generated Data

In this section, we further investigate the effectiveness of reflection learning on the reflector R by adopting self-training on **ReflectEvo-460k** using supervised fine-tuning (SFT) (Ouyang et al., 2022) and Direct Preference Optimization (DPO) (Rafailov et al., 2024).

3.1 Reflection Learning

We use **SFT** on $\mathcal{D}^+$ in two different settings below. This strengthens the model to better leverage reflections as intermediate thoughts leading to positive $\hat{a}$ for refinement.

Setting 1: We train the capability of self-reflection and self-correction in one stage:

$$\mathcal{L}_1 = -\mathbb{E}_{(q,a,f,r,\hat{a})\sim\mathcal{D}^+} \log R((r,\hat{a}) \mid q,a,f) \quad (5)$$

Setting 2: We train the capacity of self-reflection and self-correction respectively in two stages:

$$\mathcal{L}_{2.1} = -\mathbb{E}_{(q,a,f,r)\sim\mathcal{D}^+} \log R(r \mid q,a,f) \quad (6)$$

$$\mathcal{L}_{2.2} = -\mathbb{E}_{(q,a,f,r,\hat{a})\sim\mathcal{D}^+} \log R(\hat{a} \mid q,a,f,r) \quad (7)$$

Inspired by error-driven learning from humans, we also leverage negative samples $\hat{a}^-$ that comprise a large portion of $\mathcal{D}^{\pm}$ and offer valuable insights for model enhancement. In addition, we assume GPT-4o with better self-reflection, which is required for reflection preference annotation as $\mathcal{D}^{\text{pref}}$, guiding SLMs to continuously refine reflections. We use preference learning through **DPO** on the aforementioned pairwise data to better judge and distinguish high-quality reflections from suboptimal ones in the following settings.

Setting 3: We train self-reflection only on $\mathcal{D}^{\pm}$:

$$\mathcal{L}_3 = -\mathbb{E}_{(x,r^+,r^-)\sim\mathcal{D}^{\pm}} \log \sigma[r_\theta(x,r^+) - r_\theta(x,r^-)] \quad (8)$$

$$r_\theta(x,r) = \beta \log \frac{\pi_\theta(r \mid x)}{\pi_{\text{ref}}(r \mid x)}, x = (q,a,f) \quad (9)$$

Setting 4: We train self-reflection only on $\mathcal{D}^{\text{pref}}$:

$$\mathcal{L}_4 = -\mathbb{E}_{(x,r^{\text{cho}},r^{\text{rej}})\sim\mathcal{D}^{\text{pref}}} \log \sigma[r_\theta(x,r^{\text{cho}}) - r_\theta(x,r^{\text{rej}})] \quad (10)$$

where $R(\cdot)$ is the policy model π_θ and $G(\cdot)$ is the reference model π_{ref}. σ is the logistic function and β is a hyperparameter that controls the proximity to the reference policy $G(\cdot)$ in both settings 3 and 4. The objective is to steer $R(\cdot)$ towards increasing the likelihood of r^+ with the correct solutions $\hat{a}$ or chosen r^{cho} and decreasing the likelihood of r^- with incorrect solutions $\hat{a}$ or rejected r^{rej} for given (q,a,f). More details can be found in Appendix B.2.

3.2 Inference

During inference, the process follows the same steps as those of reflection data generation in Sec. 2.2. We use the model after reflection learning as a reflector at the inference time for self-reflection and correction. It can be implemented as a multi-turn rollout that terminates either when the current a is judged to be correct or when it reaches the predefined maximum number of turns (two turns in our setting using twice QA with one intermediate reflection).

4 Experiments

4.1 Performance Learning on ReflectEvo

We measure the performance of self-reflection by adopting the following metrics: 1) **Acc@t1**: the model's accuracy in the first turn; 2) **Acc@t2**: the model's accuracy in the second turn; 3) **Δ(t1,t2)**: accuracy improvement between the first and second turns measuring the efficacy of self-reflection.

We compare three main methods in our experiments, including prompt-based QA with or without reflection without training, SFT training with direct answers, and self-training based reflection learning introduced in Sec. 3 from Setting 1 to Setting 4, noted as one-stage w/ $\mathcal{D}^{+}$, two-stage w/ $\mathcal{D}^{+}$, w/ $\mathcal{D}^{\pm}$ and w/ $\mathcal{D}^{\text{pref}}$ respectively.

Overall Performance on Different Tasks Tab. 2 illustrates the overall performance on ReflectEvo. We discard the self-generated reflection data by Mistral on MATH due to its extremely low quality. We observe that LLMs gain more from prompt-based reflection, whereas SLMs show either minor improvements or degradation. This is primarily because without specialized training, SLMs inherently generate low-quality reflections and fail to leverage feedback effectively for self-correction. For comparison, experiments on our self-training methods show significant improvements in both models and various tasks. Specifically, it achieves over 20% in Δ(t1,t2) for Llama-3 on MBPP and BIG-bench as well as Mistral on LogiQA and BIG-bench. Notably, all three evaluated models outperform their stronger model using ReflectEvo on BIG-bench. This indicates that different models and tasks benefit greatly from the four self-training methods, even surpassing the SFT on answers without step-by-step reasoning process, which paves the way for broader applications and scenarios for various SLMs.

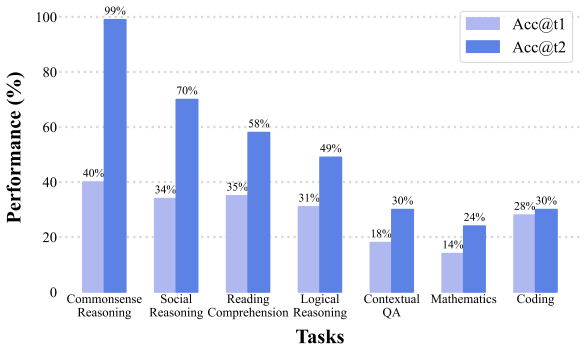


Figure 3: Performance training with ReflectEvo across different tasks on Llama-3-8B.

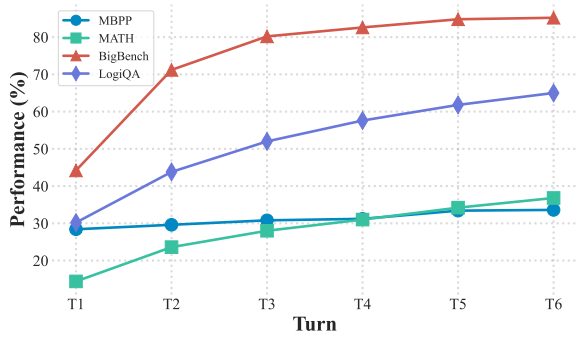


Figure 4: Performance in multi-turn self-reflection on Llama-3-8B after tuning.

Fig. 3 provides an in-depth analysis on the reflection learning across tasks. Our method significantly contributes to various tasks, including reasoning, math, QA, and comprehension, with an average of 22% in Δ(t1,t2). For coding, it only improves to a certain degree probably due to the lack of fine-grained step-by-step critiques on the erroneous solutions for reflection training on models that are not specialized in coding.

Effect of Reflection from Teacher Model To investigate the influence of reflection sources, we compare different reflections generated by the SLM itself and a more advanced model like GPT-4o which acts as a teacher model with greater knowledge and reasoning capabilities in Tab. 3. Reflections from both models strengthen the Δ(t1,t2) of QA performance after tuning under different settings proposed in Sec. 3, while the self-generated data require less cost and resources in practice. To our expectation, reflections from the teacher model yields more obvious improvements underscoring the benefits of high-quality reflection data generation and selection for further improvement.

Scaling Multi-turn Self-reflection We further extend the application of self-reflection to multi-

	LogiQA			MATH			MBPP			BIG-bench		
	Acc@t1	Acc@t2	Δ(t1,t2)	Acc@t1	Acc@t2	Δ(t1,t2)	Acc@t1	Acc@t2	Δ(t1,t2)	Acc@t1	Acc@t2	Δ(t1,t2)
Meta-Llama-3-8B-Instruct												
Prompt based												
↪ w/o reflection	30.2%	38.8%	+8.6%	14.4%	15.0%	+0.6%	28.4%	44.0%	+15.6%	38.2%	52.4%	+14.2%
↪ w/ reflection		36.2%	+6.0%		16.0%	+1.6%		45.8%	+17.4%		51.0%	+12.8%
SFT based												
↪ w/ SFT qa pairs	46.6%	-	-	10%	-	-	31.2%	-	-	61.6%	-	-
Self-training based (Ours)												
↪ one-stage w/ $\mathcal{D}^+$	30.2%	**43.8%**	**+13.6%**	14.4%	**23.6%**	**+9.2%**	28.4%	29.6%	+1.2%	38.2%	**71.2%**	**+33.0%**
↪ two-stage w/ $\mathcal{D}^+$		**49.4%**	**+19.2%**		14.5%	+0.1%		42.4%	+14.0%		45.4%	+7.2%
↪ w/ $\mathcal{D}^{\pm}$		**41.8%**	**+11.6%**		15.2%	+0.8%		**48.8%**	**+20.4%**		**63.0%**	**+24.8%**
↪ w/ $\mathcal{D}^{\text{pref}}$		**39.2%**	**+9.0%**		14.8%	+0.4%		**47.4%**	**+19.0%**		**59.6%**	**+21.4%**
Meta-Llama-3-70B-Instruct												
↪ w/o reflection	42.4%	64.4%	+22.0%	40.8%	49.6%	+8.8%	66.2%	71.0%	+4.8%	48.0%	67.0%	+19.0%
↪ w/ reflection		56.8%	+14.4%		48.6%	+7.8%		73.0%	+6.8%		64.4%	+16.4%
Mistral-7B-Instruct-v0.2												
Prompt based												
↪ w/o reflection	28.8%	31.2%	+2.4%	9.2%	10.6%	+1.4%	20.4%	23.0%	+2.6%	36.6%	43.8%	+7.2%
↪ w/ reflection		34.2%	+5.4%		10.2%	+1.0%		23.6%	+3.2%		44.4%	+7.8%
SFT based												
↪ w/ SFT qa pairs	28.8%	-	-	7.6%	-	-	17.0%	-	-	37.8%	-	-
Self-training based (Ours)												
↪ one-stage w $\mathcal{D}^+$	28.8%	**38.4%**	**+9.6%**	-	-	-	20.4%	**24.0%**	**+3.6%**	36.6%	**51.6%**	**+15.0%**
↪ two-stage w/ $\mathcal{D}^+$		**48.8%**	**+20.0%**	-	-	-		20.8%	+0.4%		**71.1%**	**+34.5%**
↪ w/ $\mathcal{D}^{\pm}$		**39.2%**	**+10.4%**	-	-	-		23.2%	+2.8%		**50.2%**	**+13.6%**
↪ w/ $\mathcal{D}^{\text{pref}}$		**38.0%**	**+9.2%**	-	-	-		22.6%	+2.2%		**48.4%**	**+11.8%**
Mistral-22B-Small-Instruct												
↪ w/o reflection	46.4%	62.8%	+16.4%	47.4%	56.2%	+8.8%	63.0%	68.0%	+5.0%	54.4%	67.2%	+12.8%
↪ w/ reflection		62.0%	+15.6%		52.8%	+5.4%		68.0%	+5.0%		68.0%	+13.6%

Table 2: Performance on Llama-3 and Mistral using ReflectEvo.

Dataset	Method	Acc@t2	Δ(t1,t2)
SR	prompt-based	36.2%	+6.0%
	one stage w/ $\mathcal{D}^+$	43.8%	+13.6%
	two stage w/ $\mathcal{D}^+$	**49.4%**	**+19.2%**
	w/ $\mathcal{D}^{\pm}$	41.8%	+11.6%
	w/ $\mathcal{D}^{\text{pref}}$	39.2%	+9.0%
TR	prompt-based	46.2%	+16.0%
	one stage w/ $\mathcal{D}^+$	**52.0%**	**+21.8%**
	two stage w/ $\mathcal{D}^+$	41.2%	+11.0%
	w/ $\mathcal{D}^{\pm}$	48.8%	+18.6%
	w/ $\mathcal{D}^{\text{pref}}$	48.0%	+17.8%

Table 3: Performance on LogiQA using different sources of reflections on Llama-3 (Acc@t1=30.2% from Tab. 2). **SR** and **TR** indicate self-reflection and teacher-reflection respectively.

turn QA in Fig. 4. To our expectation, the results demonstrate a consistent improvement with increasing turns of reflection on different tasks. BIG-bench exhibits the most significant improvement, surpassing 80% accuracy after six turns and LogiQA also shows a notable upward trend, highlighting the effectiveness of iterative refinement. MBPP and MATH display relatively modest improvements with gradual increase, which suggesting that the impact of self-reflection learning is broadly beneficial but varies between tasks. It is encouraged to investigate the underlying factors that contribute to these differences to further enhance performance in various tasks.

Generalization across Different Tasks and Models We conduct deeper studies on the generalization of the self-reflection capability after tuning across different tasks (Tab. 4) and models (Tab. 5). Our findings reveal that the benefits of reflection learning generalize across tasks, particularly for LogiQA and BIG-bench with 10% increase, which commonly require strong reasoning abilities from LLMs. Due to the divergence of MATH and MBPP, there is merely improvement when trained on reflections generated from the other three datasets. We observe that all the test models in Tab. 5 benefit from the reflector after tuning for error correction in Acc@t2, especially for initial solutions from different generators. For Mistral and Gemma, even with a minor decrease compared with the corresponding results in Tab. 2 and Tab. 9, the result on these two models highlights the potential of our pipeline across different models and demonstrates the effectiveness of reflectors when applied to various generators.

	LogiQA			MATH			BIG-bench			MBPP		
	Acc@t1	Acc@t2	Δ(t1,t2)	Acc@t1	Acc@t2	Δ(t1,t2)	Acc@t1	Acc@t2	Δ(t1,t2)	Acc@t1	Acc@t2	Δ(t1,t2)
Prompt based												
↪ w/ reflection	30.2%	36.2%	+6.0%	14.4%	16.0%	+1.6%	38.2%	51.0%	+12.8%	28.4%	45.8%	+17.4%
Self-training based (Ours)												
↪ w/ $\mathcal{D}^+$ on LogiQA	30.2%	-	-	14.4%	14.4%	+0.0%	38.2%	**60.0%**	**+21.8%**	28.4%	30.6%	+2.2%
↪ w/ $\mathcal{D}^+$ on MATH		36.6%	+6.4%		-	-		**54.8%**	**+16.6%**		28.8%	+0.4%
↪ w/ $\mathcal{D}^+$ on BIG-bench		**52.0%**	**+21.8%**		14.4%	+0.0%		-	-		36.2%	+7.8%
↪ w/ $\mathcal{D}^+$ on MBPP		30.4%	+0.2%		14.6%	+0.2%		40.2%	+2.0%		-	-

Table 4: Generalization across tasks for Llama-3 training one-stage with different task-specific subsets in $\mathcal{D}^+$.

Different generators	Acc@t1	Acc@t2	Δ(t1,t2)
Mistral-7B	28.8%	45.8%	+17.0%
Gemma-2-9B	47.6%	57.2%	+9.6%
Llama-3.1-8B	37.4%	51.0%	+13.6%

Table 5: Generalization across generators using the same reflector Llama-3 training one-stage with subset of LogiQA in $\mathcal{D}^+$.

Self-training method	Acc@t1	Acc@t2	Δ(t1,t2)
one-stage w/ $\mathcal{D}^+$	28.8%	**40.2%**	**+11.4%**
two-stage w/ $\mathcal{D}^+$	28.8%	38.0%	+9.2%
w/ $\mathcal{D}^{\pm}$	28.8%	39.4%	+10.6%
w/ $\mathcal{D}^{\text{pref}}$	28.8%	38.6%	+9.8%

Table 6: Generalization of generated reflection data on LogiQA of LlaMA-3.1-8B training on Mistral-7B

In Tab. 6, we further explore whether the self-reflection data of one LLM can be beneficial for the other. Compared with Tab. 2 in our paper, we find that the reflection data generated by LlaMA-3.1-8B is helpful for Mistral-7B on reflection learning with comparable or even better performance. It indicates that our dataset **ReflectEvo-460k** could be reusable for the community for future studies.

Effect of Different Verifiers on Self-Reflection In this paper, self-reflection is performed only when the model's answer is verified as incorrect using the ground truth. Another potential approach is to train the model itself as a verifier or use an external reward function to score the model's answer based on a predefined threshold. We compare the effects of using oracle ground truth and self-judgments generated by the baseline model as verifiers in Tab. 7. For both verifiers, reflection learning improves Acc@t1 by an average of 13+% and enhances Acc@t2 by up to 7% compared with the untuned version. Although the baseline model, without specialized training, exhibits occasional misjudgments, its verification process results in minor performance degradation on the advantage of reflection learning. We leave this a direction for further exploration on the optimized verifiers in an end-to-end pipeline.

Self-training Method	Oracle Groundtruth	Self-judgement
one-stage w/ $\mathcal{D}^+$	43.8%	32.8%
two-stage w/ $\mathcal{D}^+$	49.4%	50.2%
w/ $\mathcal{D}^{\pm}$	41.8%	40.6%
w/ $\mathcal{D}^{\text{pref}}$	39.2%	37.8%

Table 7: Acc@t2 using different verifiers during inference on LogiQA for Llama-3. (Acc@t1=30.2% and Acc@t2= 36.2% without tuning from Tab. 2)

Methods	Acc@t1	Acc@t2	Δ(t1,t2)
STaR (Zelikman et al., 2022)	40.0%	-	-
Re-ReST (Dou et al., 2024)	38.8%	-	-
RISE (Qu et al., 2024)	31.4%	34.4%	+3.0%
one-stage w/ $\mathcal{D}^+$	30.2%	43.8%	+13.6%
two-stage w/ $\mathcal{D}^+$	30.2%	**49.4%**	**+19.2%**
w/ $\mathcal{D}^{\pm}$	30.2%	41.8%	+11.6%
w/ $\mathcal{D}^{\text{pref}}$	30.2%	39.2%	+9.0%

Table 8: Performance on different baselines using LlaMA-3.1-8B on LogiQA.

In Tab. 8, we make further experiments by comparing with three well-acknowledged baselines for self-improvement through reflection or correction. To make the comparison fair, the external feedback used in all the experiments is only a binary signal "correct/incorrect" without further explanation or given ground truth. We follow the evaluation setting in the original paper. Comparing with STaR and Re-ReST, Ours benefit from improvement over turns. Our methods achieves much higher reasoning performance after self-reflection, which emulates human learning through a slower and deeper thought process that iteratively and ultimately derives self-evolution

4.2 In-depth Analysis on Reflection

Error Types Identified by Reflection To dissect the intrinsic properties of our reflection data, we analyze the error types in the initial thoughts specified by the reflection across all test sets. Potential error labels are generated heuristically by auto-tagging with GPT-4o and then calibrated by human annotators, achieving Cohen's kappa of 51.18% with moderate agreement (Landis and Koch, 1977) indi-

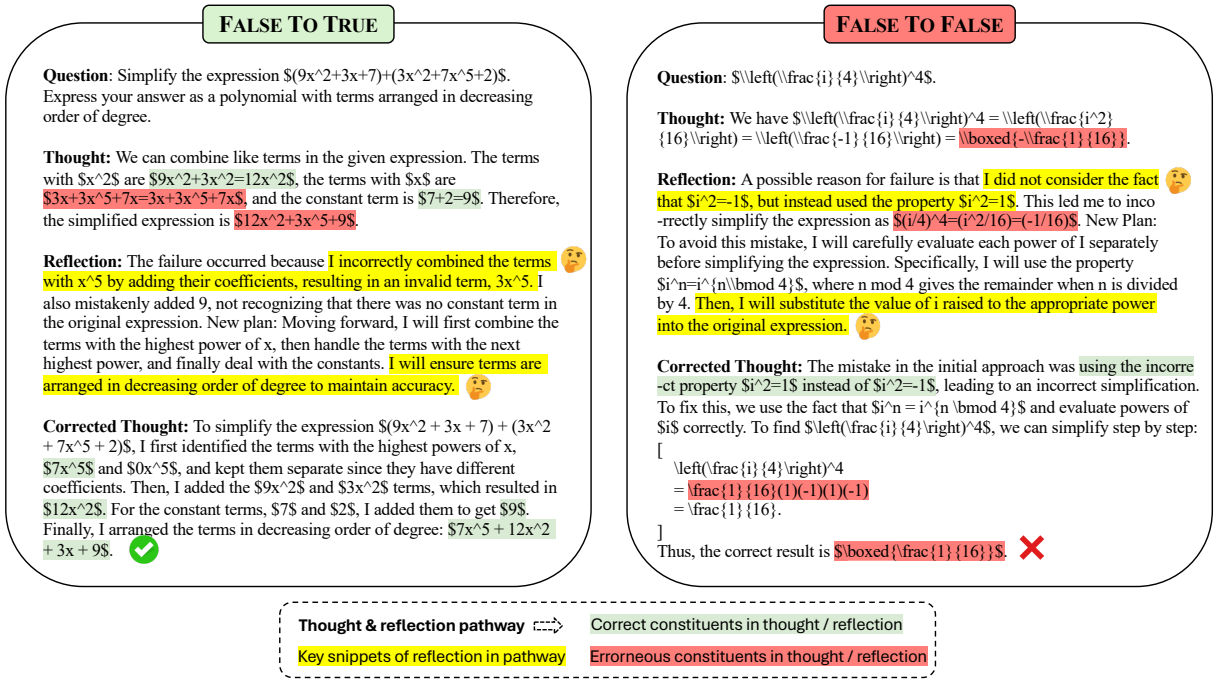


Figure 5: Qualitative examples from the MATH. "False to True" and a "False to False" stand for successful and failed correction in the second turn respectively. The key snippets highlighted in green, red and yellow indicate correct, erroneous thought and reflection respectively.

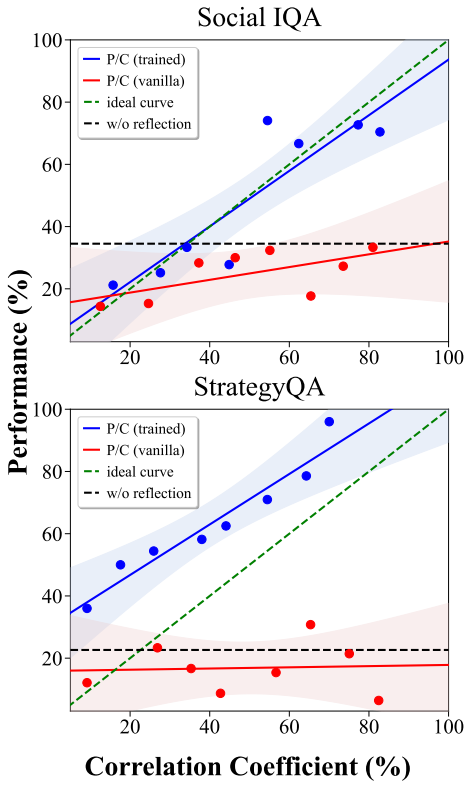


Figure 6: Task performance (Acc@t2) versus the correlation between reflection and the second-turn thought.

cating high annotation quality.

Fig. 2b shows five coarse-grained and nine fine-grained error types identified through human calibration. The most common errors are *Logic and Reasoning Errors* (88.4%) and *Instruction Violation* (47.9%), indicating that math and logic issues were the primary causes. We also provide detailed error distributions for the different subsets. MATH has a higher percentage of *Calculation Errors* (20.8%) than the other subsets, whereas COQA has more *Context misinterpretation* (43.1%). This shows that our method provides tailored reflections for specific domains rather than superficial or general advice.

Correlation Analysis in Reflection We calculate the correlation between reflection and second-turn corrected thoughts, and we assess the association between the correlation and Acc@t2 after self-correction. Empirically, we hypothesize that they have a linear relationship, and we select the Pearson correlation coefficient by computing the semantic similarity for each pair of data (see the details in Fig. 7). As we have seen, reflection learning can improve the ability of models to correct errors; we argue that if reflection is indeed specific to the error in thought, then task performance should intuitively be enhanced as the correlation between reflection and corrected thought increases.

Measuring with the Pearson coefficient, Fig. 6 and Fig. 7 show that StrategyQA, Social IQa, VitaminC, and SQuAD all have a clear linear relationship between the performance and the correlation of reflection – second-turn thought, while MATH

and MBPP exhibit irrelevant tendency or show a slightly negative correlation implying their desire data of fine-grained reflection. Comparing the blue and red correlation curves, we find that more similarity between the reflection and corrected thought, more effective correction (*i.e.*, higher performance) that outperforms the vanilla model.

Case Studies We perform case studies to see how reflection interacts with the thought process by making critiques and refinements in Fig. 5. We random sampled 100 cases from the MATH test set and display two of them. In the case "False to True", reflection precisely recaps the key causes of error and explicitly bridges the logical pathway between the initial thought and the corrected one, which finally results in the correct answer. In contrast, we find that even tiny erroneous constituent in the reflection may lead to a false reasoning thought and final answer. It validates that high-quality reflection is helpful for incentivizing the model to generate thought with correct answer while flawed reflection still lead to repeated errors after self-correction.

5 Related Work

Self-training and Self-Improvement Self-training allows a model to learn from its own outputs, reducing its reliance on human-annotated data or superior models (Zelikman et al., 2022; Yuan et al., 2024; Chen et al., 2024). Previous research has primarily concentrated on enhancing models' reasoning abilities through SFT (Yuan et al., 2023) with positive samples or preference learning using both positive and negative samples to potentially leverage valuable information in incorrect solutions and recent advances also extend self-training to agentic scenarios (Wang et al., 2024a; Wallace et al., 2024; Gulcehre et al., 2023; Song et al., 2024; Motwani et al., 2024; Li et al., 2024b). We further advocate reducing reliance on resource-heavy rationale annotations via self-training for SLMs.

Learning for Self-reflection Recent research highlights the significant benefits of integrating self-reflection into LLMs to enhance their reasoning and problem-solving capabilities, by iteratively refining their responses (Kumar et al., 2024; Cheng et al., 2024; Qu et al., 2024; Yao et al., 2023; Zhou et al., 2024; Liang et al., 2024; Moskvoretskii et al., 2025). Shinn et al. (2024) reinforces the language agent to verbally reflect on task feedback and induce better plans in subsequent trials. Dou et al. (2024) employs the low-quality outputs generated from the weak model iteratively by fine-tuning the reflection module for self-refinement. Zhang et al. (2024) further validates that SLMs have the ability of self-correction on reasoning tasks by accumulating high-quality critique-correction data. We pioneer the exploration of reflection learning on self-generated data.

6 Conclusion

We propose ReflectEvo to enhance SLMs through reflection learning by iteratively generating self-training data, which achieves substantial performance improvements, even surpassing much larger models highlighting its generalization across various models and tasks for future research.

Limitations

Despite the promising results of **ReflectEvo** through reflection learning, there are several limitations to our work. The quality of the self-generated reflection data is highly dependent on the initial reasoning ability of the SLMs. Models with inherently weak reasoning capabilities may struggle to produce high-quality reflections, which in turn limits the effectiveness of the self-training. While our pipeline demonstrates significant improvements in certain tasks, tasks such as coding and mathematics require more specialized knowledge and step-by-step critiques than reasoning and comprehension tasks. Future work could explore more sophisticated feedback mechanisms with optimized verifiers or reward functions during inference to enhance the reflection learning process.

Acknowledgement

The authors thank the reviewers for their insightful suggestions on improving the manuscript. This work presented herein is supported by the National Natural Science Foundation of China (62376031).

Ethics Statement

We adhere to ethical principles to ensure the responsible development and application of our proposed techniques. Our work focuses on enhancing the self-reflection abilities of SLMs without directly involving human subjects or sensitive information. We acknowledge the potential broader impacts of our research, recognize the environmental and computational costs associated with LLM training, and strive to optimize our methods for efficiency.

References

Sue Atkins and Kathy Murphy. 1993. Reflection: a review of the literature. *Journal of advanced nursing*, 18(8):1188–1192.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. 2021. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*.

BIG bench authors. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*.

Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. 2024. Self-play fine-tuning converts weak language models to strong language models. *arXiv preprint arXiv:2401.01335*.

Kanzhi Cheng, Yantao Li, Fangzhi Xu, Jianbing Zhang, Hao Zhou, and Yang Liu. 2024. Vision-language models can self-improve reasoning via reflection. *arXiv preprint arXiv:2411.00855*.

David Denton. 2011. Reflection and learning: Characteristics, obstacles, and implications. *Educational Philosophy and Theory*, 43(8):838–852.

Zi-Yi Dou, Cheng-Fu Yang, Xueqing Wu, Kai-Wei Chang, and Nanyun Peng. 2024. Re-rest: Reflection-reinforced self-training for language agents. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 15394–15411.

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

Thomas Palmeira Ferraz, Kartik Mehta, Yu-Hsiang Lin, Haw-Shiuan Chang, Shereen Oraby, Sijia Liu, Vivek Subramanian, Tagyoung Chung, Mohit Bansal, and Nanyun Peng. 2024. Llm self-correction with decrim: Decompose, critique, and refine for enhanced following of instructions with multiple constraints. *arXiv preprint arXiv:2410.06458*.

Caglar Gulcehre, Tom Le Paine, Srivatsan Srinivasan, Ksenia Konyushkova, Lotte Weerts, Abhishek Sharma, Aditya Siddhant, Alex Ahern, Miaosen Wang, Chenjie Gu, et al. 2023. Reinforced self-training (rest) for language modeling. *arXiv preprint arXiv:2308.08998*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Jiayi He, Hehai Lin, Qingyun Wang, Yi Fung, and Heng Ji. 2024. Self-correction is more than refinement: A learning framework for visual and language reasoning tasks. *arXiv preprint arXiv:2410.04055*.

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

Jiaxin Huang, Shixiang Shane Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. 2022. Large language models can self-improve. *arXiv preprint arXiv:2210.11610*.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Aviral Kumar, Vincent Zhuang, Rishabh Agarwal, Yi Su, John D Co-Reyes, Avi Singh, Kate Baumli, Shariq Iqbal, Colton Bishop, Rebecca Roelofs, et al. 2024. Training language models to self-correct via reinforcement learning, 2024. *URL https://arxiv.org/abs/2409.12917*.

J Richard Landis and Gary G Koch. 1977. The measurement of observer agreement for categorical data. *biometrics*, pages 159–174.

Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. 2025. NV-embed: Improved techniques for training LLMs as generalist embedding models. In *The Thirteenth International Conference on Learning Representations*.

Jiaqi Li, Mengmeng Wang, Zilong Zheng, and Muhan Zhang. 2024a. Loogle: Can long-context language models understand long contexts? In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Jiaqi Li, Xiaobo Wang, Wentao Ding, Zihao Wang, Yipeng Kang, Zixia Jia, and Zilong Zheng. 2024b. Ram: Towards an ever-improving memory system by learning from communications. *arXiv preprint arXiv:2404.12045*.

Ming Li, Lichang Chen, Jiuhai Chen, Shwai He, Heng Huang, Jiuxiang Gu, and Tianyi Zhou. 2023. Reflection-tuning: Data recycling improves llm instruction-tuning. *arXiv preprint arXiv:2310.11716*.

Youquan Li, Miao Zheng, Fan Yang, Guosheng Dong, Bin Cui, Weipeng Chen, Zenan Zhou, and Wentao Zhang. 2024c. Fb-bench: A fine-grained multi-task benchmark for evaluating llms' responsiveness to human feedback. *arXiv preprint arXiv:2410.09412*.

Yiming Liang, Ge Zhang, Xingwei Qu, Tianyu Zheng, Jiawei Guo, Xinrun Du, Zhenzhu Yang, Jiaheng Liu, Chenghua Lin, Lei Ma, et al. 2024. I-sheep: Self-alignment of llm from scratch through an iterative self-enhancement paradigm. *arXiv preprint arXiv:2408.08072*.

Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. 2020. Logiqa: A challenge dataset for machine reading comprehension with logical reasoning. *arXiv preprint arXiv:2007.08124*.

Tianqi Liu, Yao Zhao, Rishabh Joshi, Misha Khalman, Mohammad Saleh, Peter J Liu, and Jialu Liu. 2023. Statistical rejection sampling improves preference optimization. *arXiv preprint arXiv:2309.06657*.

Trung Quoc Luong, Xinbo Zhang, Zhanming Jie, Peng Sun, Xiaoran Jin, and Hang Li. 2024. Reft: Reasoning with reinforced fine-tuning. *arXiv preprint arXiv:2401.08967*.

Viktor Moskvoretskii, Chris Biemann, and Irina Nikishina. 2025. Self-taught self-correction for small language models. *arXiv preprint arXiv:2503.08681*.

Sumeet Ramesh Motwani, Chandler Smith, Rocktim Jyoti Das, Markian Rybchuk, Philip HS Torr, Ivan Laptev, Fabio Pizzati, Ronald Clark, and Christian Schroeder de Witt. 2024. Malt: Improving reasoning with multi-agent llm training. *arXiv preprint arXiv:2412.01928*.

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.

Jing-Cheng Pang, Pengyuan Wang, Kaiyuan Li, Xiong-Hui Chen, Jiacheng Xu, Zongzhang Zhang, and Yang Yu. 2023. Language model self-improvement by reinforcement learning contemplation. *arXiv preprint arXiv:2305.14483*.

Yuxiao Qu, Tianjun Zhang, Naman Garg, and Aviral Kumar. 2024. Recursive introspection: Teaching language model agents how to self-improve. *arXiv preprint arXiv:2407.18219*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.

Matthew Renze and Erhan Guven. 2024. Self-reflection in llm agents: Effects on problem-solving performance. *arXiv preprint arXiv:2405.06682*.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2024. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36.

Yifan Song, Da Yin, Xiang Yue, Jie Huang, Sujian Li, and Bill Yuchen Lin. 2024. Trial and error: Exploration-based trajectory optimization for llm agents. *arXiv preprint arXiv:2403.02502*.

Xiaojuan Tang, Jiaqi Li, Yitao Liang, Muhan Zhang, and Zilong Zheng. 2024. Mars: Situated inductive reasoning in an open-world environment. In *38th Conference on Neural Information Processing Systems (NeurIPS 2024) Track on Datasets and Benchmarks*.

Xiaojuan Tang, Zilong Zheng, Jiaqi Li, Fanxu Meng, Song-Chun Zhu, Yitao Liang, and Muhan Zhang. 2023. Large language models are in-context semantic reasoners rather than symbolic reasoners. *arXiv preprint arXiv: 2305.14825*.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

Zilong Zheng Tong Wu, Yanpeng Zhao. 2024. An efficient recipe for long context extension via middle-focused positional encoding. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37.

Johan Von Wright. 1992. Reflections on reflection. *Learning and instruction*, 2(1):59–68.

Bram Wallace, Meihua Dang, Rafael Rafailov, Linqi Zhou, Aaron Lou, Senthil Purushwalkam, Stefano Ermon, Caiming Xiong, Shafiq Joty, and Nikhil Naik. 2024. Diffusion model alignment using direct preference optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8228–8238.

Tianduo Wang, Shichen Li, and Wei Lu. 2024a. Self-training with direct preference optimization improves chain-of-thought reasoning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11917–11928. Association for Computational Linguistics.

Yuxuan Wang, Alan Yuille, Zhuowan Li, and Zheng Zilong. 2024b. Exovip: Step-by-step verification and exploration with exoskeleton modules for compositional visual reasoning. In *The first Conference on Language Modeling (CoLM)*.

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*.

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*.

Weiran Yao, Shelby Heinecke, Juan Carlos Niebles, Zhiwei Liu, Yihao Feng, Le Xue, Rithesh Murthy, Zeyuan Chen, Jianguo Zhang, Devansh Arpit, et al. 2023. Retroformer: Retrospective large language agents with policy gradient optimization. *arXiv preprint arXiv:2308.02151*.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. 2024. Self-rewarding language models. *arXiv preprint arXiv:2401.10020*.

Zheng Yuan, Hongyi Yuan, Chengpeng Li, Guanting Dong, Keming Lu, Chuanqi Tan, Chang Zhou, and Jingren Zhou. 2023. Scaling relationship on learning mathematical reasoning with large language models. *arXiv preprint arXiv:2308.01825*.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. 2022. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488.

Zhongshen Zeng, Yinhong Liu, Yingjia Wan, Jingyao Li, Pengguang Chen, Jianbo Dai, Yuxuan Yao, Rongwu Xu, Zehan Qi, Wanru Zhao, et al. 2024. Mr-ben: A comprehensive meta-reasoning benchmark for large language models. *arXiv preprint arXiv:2406.13975*.

Yunxiang Zhang, Muhammad Khalifa, Lajanugen Logeswaran, Jaekyeom Kim, Moontae Lee, Honglak Lee, and Lu Wang. 2024. Small language models need strong verifiers to self-correct reasoning. *arXiv preprint arXiv:2404.17140*.

Runlong Zhou, Simon S Du, and Beibin Li. 2024. Reflect-rl: Two-player online rl fine-tuning for lms. *arXiv preprint arXiv:2402.12621*.

	LogiQA			MATH			MBPP			BIG-bench		
	Acc@t1	**Acc@t2**	**Δ(t1,t2)**	**Acc@t1**	**Acc@t2**	**Δ(t1,t2)**	**Acc@t1**	**Acc@t2**	**Δ(t1,t2)**	**Acc@t1**	**Acc@t2**	**Δ(t1,t2)**
Gemma-2-9B-it												
Prompt based												
↪ w/o reflection	47.6%	63.0%	+15.4%	34.6%	40.4%	+5.8%	54.4%	59.2%	+4.8%	61.2%	75.6%	+14.4%
↪ w/ reflection		60.0%	+12.4%		40.0%	+5.4%		59.6%	+5.2%		74.4%	+13.2%
SFT based												
↪ w/ SFT qa pairs	50.6%	-	-	18.6%	-	-	37.0%	-	-	74.6%	-	-
Self-training based (Ours)												
↪ one-stage w/ D^+	47.6%	62.4%	+14.8%	34.6%	40.0%	+5.4%	54.4%	57.8%	+3.4%	61.2%	**78.4%**	**+17.2%**
↪ two-stage w/ D^+		60.6%	+13.0%		35.0%	+0.4%		56.8%	+2.4%		67.0%	+5.8%
↪ w/ $D^{\pm}$		62.6%	+15.0%		40.0%	+5.4%		58.6%	+4.2%		74.8%	+13.6%
↪ w/ D^{pref}		63.0%	+15.4%		40.0%	+5.4%		59.4%	+5.0%		75.0%	+13.8%
Gemma-2-27B-it												
↪ w/o reflection	52.0%	59.2%	+7.2%	38.4%	43.8%	+5.4%	65.4%	69.4%	+4.0%	63.4%	72.0%	+8.6%
↪ w/ reflection		65.4%	+13.4%		45.0%	+6.6%		69.2%	+3.8%		75.2%	+11.8%

Table 9: Performance on Gemma-2 using ReflectEvo.

A Further analyais and results

Tab. 9 indicates that BIG-bench gains more from reflection tuning with Acc@t2 = 78.4% with a substantial increase of +17.2% compared with other tasks and the baseline methods. However, our method on Gemma-2 shows marginal improvement compared with Llama-3 and Mistral probably due to its inherent strong reasoning capability (comparable performance on different models in sizes of 9B & 27B). It may either need selection of higher-quality reflection data or supervision from superior models and further optimization on the training methods for reflection enhancement. Due to the defect of SFT training without step-by-step reasoning process and the limited number of training data, the SFT performance of MATH and MBPP degrade due to the nature of fast thinking than thoses with reflection.

B Implementation Details

B.1 Reflection generation

In this paper, we conduct the self-reflection once during the process of the two turns of reasoning and answering for both data generation and inference across most experiments. The generalization performance of multi-turn self-reflection can be found in Fig. 4.

The number of reject sampling k is 2. To validate the effectiveness of reflection tuning on various tasks, we incorporate 14 datasets selected from BIG-bench besides LogiQA, MATH and MBPP. Those datasets are delicately selected to focus more on the comprehension and reasoning abilities across diverse domains, comprising a comprehensive collection of dataset. The datasets includes: Commonsense Reasoning (RiddleSense, TimeDial, Known Unknowns), Social Reasoning (Social IQa, Implicit Interpersonal Relations), Reading Comprehension (VitaminC, SQuADShifts), Logical Reasoning (StrategyQA, Analytic Entailment), Contextual QA (CoQA Conversational Question Answering), Context Free QA (Truthful QA), Causal Reasoning (Causal Judgment, Cause and Effect), and Physics Reasoning (Physical Intuition). For datasets with more than 1000 samples, we randomly select 1000 QA pairs; for datasets with fewer than 1000 samples, we retain the entire original set.

Each reflection instruction consists of the three stages introduced in step 1 in Sec. 2 and different variants of prompts used in each stage can be seen Appendix C.1. The combination of them forms a diverse, comprehensive instruction pool with 32 (2*8*2) instructions used in step 2. For each dataset, we random select 5 or 6 of the instructions (M) to generate the reflections in **ReflectEvo-460k** considering the data generation efficiency.

For each task, we use corresponding subset for training without using the whole **ReflectEvo-460k**. For example, we use the training set of LogiQA for reflection generation and learning for LlaMA-3.1-8B and evaluate the same model on the test set of LogiQA in the experiments.

	LogiQA			MATH			BIG-bench			MBPP		
	D^+	$D^\pm$	D^{pref}	D^+	$D^\pm$	D^{pref}	D^+	$D^\pm$	D^{pref}	D^+	$D^\pm$	D^{pref}
Training	25371	152475	59870	13796	50946	28225	20410	70672	30909	1151	5365	2609
Testing		500			500			500			500	

Table 10: Data statistics for training and testing samples in the experiments.

Hyperparameter	one-stage w/ D^+	two-stage w/ D^+	w/ $D^\pm$	w/ D^{pref}
learning rate	1e-3	1e-5,1e-3	5e-7, 7e-7	5e-7, 7e-7
weight decay	0	0-0.01	-	-
max grad norm	1.0	1.0	-	-
β_1 for SFT	0.9	0.99	-	-
β_2 for SFT	0.999	0.9	-	-
β for DPO	-	-	0.01	0.01
ϵ	1e-8	1e-08	-	-
max new tokens	512	512	248	248

Table 11: The hyperparameters for reflection tuning.

B.2 Training

All the experiments for reflection tuning can be conducted on two Nvidia A100 80GB GPU, 32GB memory, 128 Core AMD CPU. The resource costs are mainly dependent on the tuning methods (full-parameter fine-tuning, parameter-efficient fine-tuning (PEFT) and DPO), the sizes of the models, and the sizes of the datasets. The main hyperparameters used for different settings of reflection tuning are shown in Tab. 11. The learning rate varies based on different models and tasks.

For one stage training with w/ D^+, we use LoRA-based PEFT in this setting with 4-bit quantization via BitsAndBytes. We set LoRA rank $r = 8$, scaling factor $\alpha = 32$, and a dropout rate of 0.1. The per-device batch size is set to 16 for LogiQA and 8 for others. For two stage training with w/ D^+, we use the full-parameter SFT in this setting with bfloat16 data precision. The per-device batch size is set to 16 with gradient accumulation of 4. For DPO training with both $D^\pm$ and D^{pref}, the per-device training batch size is set to 2, and gradient accumulation is set to 32.

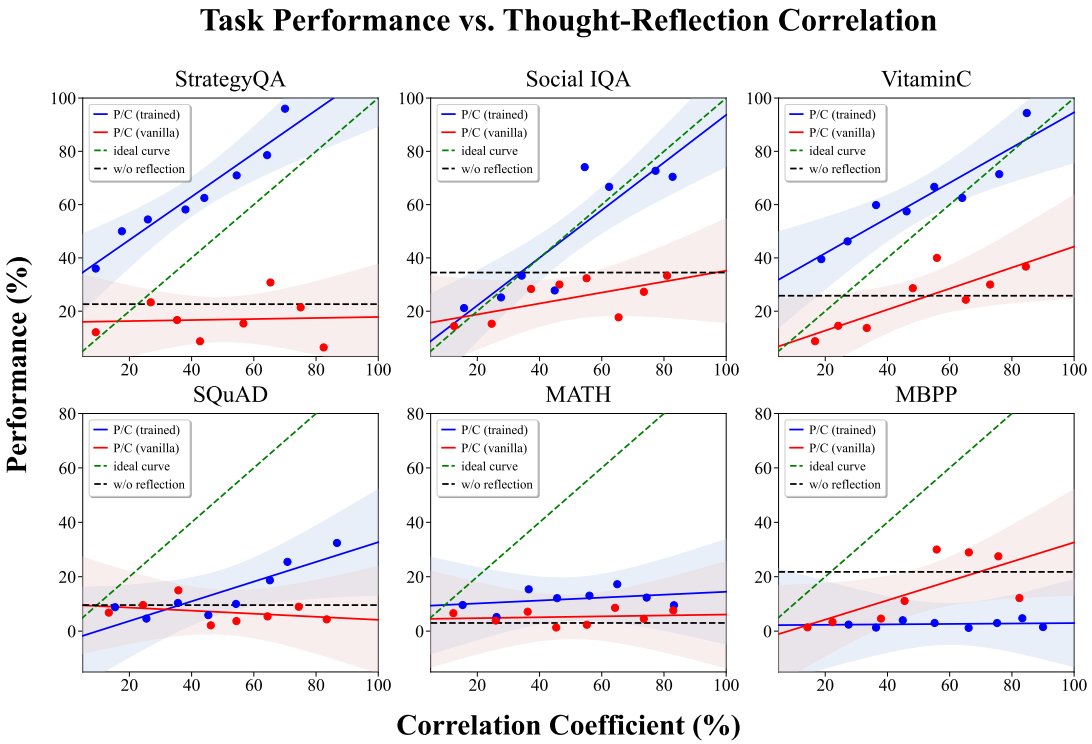


Figure 7: Task performance (Acc@t2) versus the correlation between reflection and second-turn thoughts for Llama-3-8B with self-training reflection (blue dots and curve) and prompt-based reflection (red dots and curve). The ideal correlation (green dashed curve) denotes a standard linear tendency for comparison purposes, and the black dashed line represents Llama-3-8B without reflection. Note that the y-values of the spots denote the average performance (axis-y), where an array of test data points is located in a specific interval of the correlation coefficient (axis-x), and the correlation coefficient of these spots is also averaged by the values in the same interval.

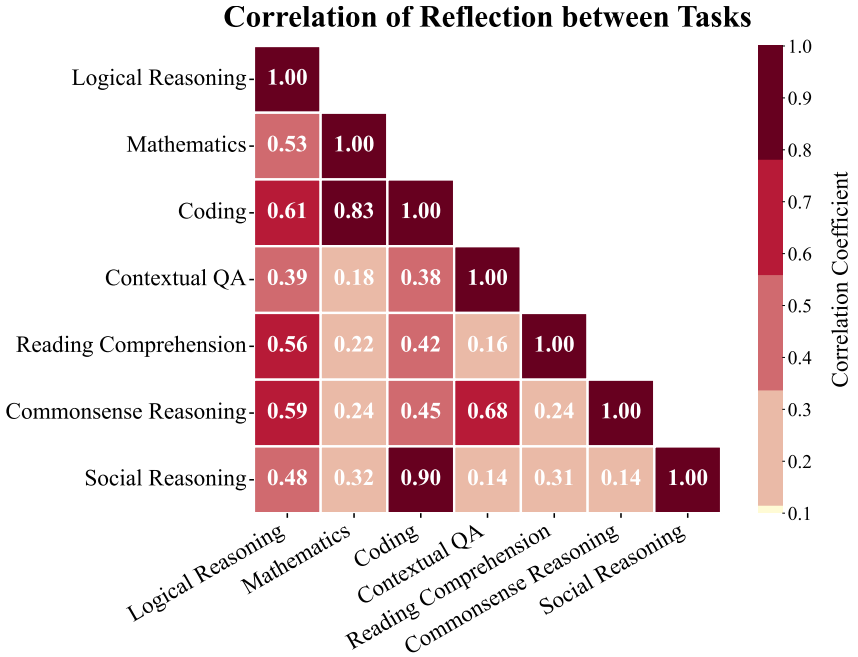


Figure 8: Correlation of reflection between each pair of tasks. We obtain the semantic representation for all reflections via the `Nv-Embed-v2` model (Lee et al., 2025) and calculate the Spearman correlation between each pair of tasks. The results are as follows: 1) Logical Reasoning has a moderate correlation with all tasks, which indicates that logic is a fundamental ability that supports other tasks; 2) Coding and Math have a high correlation, implying that similar thinking patterns are required for handling math and coding problems; and 3) Commonsense Reasoning and Social Reasoning show low correlation (0.14), suggesting that these abilities might require different cognitive skills.

C Prompts

C.1 Reflection generation for ReflectEvo-460k

Instruction: Given the question and relevant context, you were unsuccessful in answering the question. As an advanced reasoning agent, you are required to enhance your incorrect solution and correctly answer the question based on self-reflection.

Question: *{Question}*

Previous trial and your incorrect solution: *{Scratchpad}*

Based on this information, please provide the following:
Stage 1: Verify the failed solution
1-1. Analyze the failed solution by tracing and examining its execution with step-by-step verification.
1-2. Quickly go through the failed solution without step-by-step verification.

Stage 2: Locate errors and diagnose potential reasons
Identify specific steps where the errors occur and diagnose potential reasons.
2-1. Review your calculation process to ensure that all the operations are accurate.
2-2. Review your algorithm logic to ensure all steps follow the correct order.
2-3. Review your solution to ensure it maintains logical coherence.
2-4. Review your solution to check statements and conclusions for internal consistency.
2-5. Review the context and requirements presented in the question to confirm that the response aligns with them.
2-6. Review your solution to ensure that it is relevant to the question and addresses each aspect of the question.
2-7. Review your solution to ensure it conforms to the required format and guidelines in a well-organized structure.
2-8. Review your solution to ensure all provided facts are accurate and up to date.

Stage 3: Outline strategies and plans on error correction and mitigation
3-1. Outline a high-level plan explaining how these changes will mitigate similar issues.
3-2. Outline a low-level plan proposing specific strategies or corrections to address these issues.

Please follow the instructions without any additional introductory or concluding statements. Do not provide the answer directly. You will be punished to output more than 100 words.

C.2 Self-reflection for Reflector

Instruction: You are an advanced reasoning agent that can improve based on self-reflection. You will be given a previous reasoning trial in which you were given a question to answer. You were unsuccessful in answering the question. In a few sentences, diagnose a possible reason for failure and devise a new, concise, high-level plan that aims to mitigate the same failure. Use complete sentences.

Question: *{Question}*
Previous trial and your incorrect solution: *{Scratchpad}*

C.3 Reasoning for Generator

Instruction: In this task, you are required to solve a question with interleaving Thought, Action, and Observation steps. Thought allows you to reason and analyze the current situation. Action calls the 'Finish' function and fill in the answer in [ ] to finish the task after Thought. The observations will be provided to you automatically after you action.

You can think step-by-step to reach the answer. Here are some examples:
{Examples}
(END OF EXAMPLES)

You are solving the following question: *{Question}*

Below is your progress so far:
(BEGIN)
{Scratchpad}
(END)

Please complete the current step.

C.4 Self-correct for Reflector

Instruction: In this task, you are required to solve a question with interleaving Thought, Action, and Observation steps. Thought allows you to reason and analyze the current situation. Action calls the 'Finish' function and fill in the answer in [] to finish the task after Thought. The observations will be provided to you automatically after you action.

You can think step-by-step to reach the answer. Here are some examples:
{Examples}
(END OF EXAMPLES)

You are solving the following question: *{Question}*

Below is your previous reflection that helps to revise the incorrect solutions and correctly answer the question. It localizes the errors, summarizes the potential reasons for your failure and outlines a plan to mitigate the same failure:
{Reflections}

Below is your progress so far:
(BEGIN)
{Scratchpad}
(END)

Please complete the current step.

C.5 Self-reflection and correct in one stage

Instruction: You are an advanced reasoning agent that can improve based on self-reflection. You will be given a previous reasoning trial in which you were given a question to answer. You were unsuccessful in answering the question. In a few sentences, diagnose a possible reason for failure and devise a new, concise, high-level plan that aims to mitigate the same failure. Use complete sentences.

Question: *{Question}*

Previous trial and your incorrect solution: *{Scratchpad}*

Based on your self-reflection, you can think step-by-step to generate a new answer to the question. Call the 'Finish' function and fill in the answer in [ ] to finish the task.

C.6 Reasoning for direct QA for SFT

Instruction: In this task, you are required to solve a question by generating the final answer directly. Call the 'Finish' function and fill in the answer in [ ] to finish the task.
Here are some examples:
{Examples}
(END OF EXAMPLES)

You are solving the following question: *{Question}*

C.7 Reflection preference annotation by GPT

Instruction: You are a helpful assistant in evaluating the quality of reflections on an unsuccessful attempt to answer a question.

You will be provided with:

Question: *{Question}*
Groundtruth to the question: *{Answer}*
Initial student's chain of thought and answer: *{Scratchpad}*
Student A's reflection: *{Reflections 1}*
Student B's reflection: *{Reflections 2}*

Student A and Student B have both reflected on the initial student's unsuccessful attempt to answer the question. Above are their reflections that diagnose possible reasons for failure or devise a better plan to mitigate the same failure. Please determine which student's reflection is better.

Your response should be either "Student A" or "Student B" without providing any explanation or other words for your choice.

Do NOT say both/neither are good.

C.8 Heuristic Error Constituent Annotation for Reflection

Instruction: You are a professional data annotator specializing in reasoning and chain-of-thought rationale analysis. Your task is to analyze the thought process provided and identify any fine-grained labels based on the given reflection and error taxonomy.

Definitions
- Thought: The reasoning steps taken by a human or model to arrive at an answer.
- Reflection: The self-reflection of the human or model on the reasoning thought process.

Error Taxonomy
1. Mathematical Errors
• 1-1. Calculation Error
• 1-2. Algorithm Error

2. Logic and Reasoning Errors
• 2-1. Flawed Rationale Error
• 2-2. Internal Inconsistency

3. Instruction Violation
• 3-1. Context Misinterpretation
• 3-2. Incomplete or Irrelevant Response
• 3-3. Format Discrepancy

4. Factual Errors
• 4-1. Factual Errors

5. No Errors
• 5-1. No Errors Detected

Input
- Question: *{question}*
- Thought: *{thought}*
- Reflection: *{reflection}*

Output
- Labels: [Error Type(s) Assigned]
- Rationale: [Explanation for label assignment, with specific examples]