Superficial Self-Improved Reasoners Benefit from Model Merging

Xiangchi Yuan[1], **Chunhui Zhang**[2], **Zheyuan Liu**[3], **Dachuan Shi**[1], **Leyan Pan**[1], **Soroush Vosoughi**[2], **Wenke Lee**[1]
[1]Georgia Institute of Technology, [2]Dartmouth College, [3]University of Notre Dame
{xyuan300, dshi77, leyanpan, wenke}@gatech.edu
{chunhui.zhang.gr, soroush}@dartmouth.edu
zliu29@nd.edu

Abstract

As scaled language models (LMs) approach human-level reasoning capabilities, self-improvement emerges as a solution to synthesizing high-quality data corpus. While previous research has identified model collapse as a risk in self-improvement, where model outputs become increasingly deterministic, we discover a more fundamental challenge: *the superficial self-improved reasoners phenomenon*. In particular, our analysis reveals that even when LMs show improved in-domain (ID) reasoning accuracy, they actually compromise their generalized reasoning capabilities on out-of-domain (OOD) tasks due to *memorization* rather than genuine *learning*. Through a systematic investigation of LM architecture, we discover that during self-improvement, LM weight updates are concentrated in less reasoning-critical layers, leading to superficial learning. To address this, we propose Iterative Model Merging (IMM), a method that strategically combines weights from original and self-improved models to preserve generalization while incorporating genuine reasoning improvements. Our approach effectively mitigates both LM collapse and superficial learning, moving towards more stable self-improving systems. Code is available[1].

1 Introduction

The reasoning capabilities (Jaech et al., 2024; Zhang et al., 2024; Guo et al., 2025) of large language models (LLMs) largely benefits from vast amounts of high-quality reasoning data. However, as the data corpus runs out (Sutskever, 2024) and increasingly powerful models approach human-level intelligence (DeepMind, 2024a,b), pressing issues emerge: *(i)* How to advance models' reasoning capabilities despite data scarcity? *(ii)* How to obtain training data that exceeds human-level performance for next-generation models? A promising answer to both questions is model self-improvement or

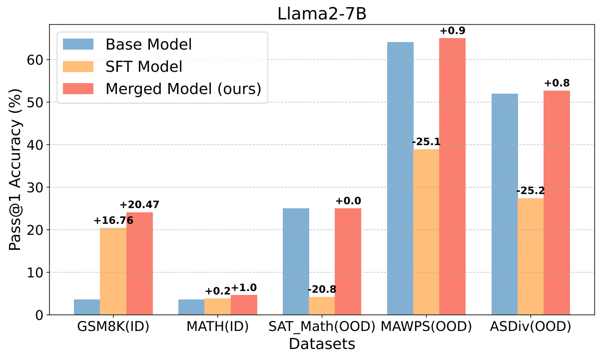

Figure 1: The Superficial Self-Improved Reasoners phenomenon is mitigated by iterative model merging. Our method improves ID and OOD reasoning performances.

self-evolution, where models autonomously generate infinite high-quality data, which potentially surpasses human annotations, to continuously enhance their own performance.

Although self-improvement has achieved remarkable success in specific domains such as mathematics (OpenAI, 2025; DeepMind, 2024a), coding (Li et al., 2022), and games (Hu et al., 2024; Silver et al., 2018), recent studies reveal significant risks associated with using self-generated synthetic data for fine-tuning: in particular, model performance can degrade over multiple iterations of self-improvement, a phenomenon known as *model collapse*. (Shumailov et al., 2023). In current research, model collapse is primarily attributed to a reduction in sampling diversity (Shumailov et al., 2023; Alemohammad et al., 2024; Guo et al., 2024b). To mitigate this problem, several studies suggest refreshing synthetic data with real data (Bertrand et al., 2024; Alemohammad et al., 2024), accumulating data across training steps (Gerstgrasser et al., 2024), and incorporating data verifiers (Gillman et al., 2024) or correctors (Feng et al., 2025). However, by focusing solely on data quality and diversity, these approaches overlook a more critical question: whether self-improvement genuinely en-

[1]Code is available at IMM.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 5901–5921

hances reasoning capabilities or merely memorizes the training distribution. This distinction becomes crucial when considering the model's ability to generalize beyond its training data.

In this paper, we investigate a risk in model self-improvement for reasoning tasks that deepens the known challenge of model collapse. We identify a phenomenon we call Superficial Self-Improved Reasoners, where models appear to improve but actually fail to develop genuine reasoning capabilities. While these models show enhanced performance on in-domain (ID) reasoning tasks, they significantly underperform on out-of-domain (OOD) tasks, suggesting memorization rather than genuine reasoning improvement. To understand the mechanistic cause of this phenomenon, we perform a systematic analysis of the model architecture during self-improvement. By examining layer importance and parameter changes, we uncover a critical mismatch: the largest weight updates occur in layers that contribute least to reasoning, while reasoning-critical layers receive minimal updates. This mismatch explains why models tend to memorize training patterns rather than develop generalizable reasoning skills. To address this issue, we propose Iterative Model Merging (IMM), a novel method that strategically combines weights from original and self-improved models. IMM specifically targets the layer misalignment problem by preserving the stability of reasoning-critical layers while allowing beneficial updates from self-improvement. As demonstrated in Figure 1, this approach effectively balances performance improvements with preserved generalized reasoning capability.

A summary of the contributions is given below:

- This work identifies the risk of self-improvement for reasoning: while the model enhances its reasoning capabilities, it still tends to memorize the training data, resulting in a loss of generalized reasoning ability. We refer to this phenomenon as *Superficial Self-Improved Reasoners*.
- We provide an explanation for this phenomenon by highlighting a mismatch between the reasoning-critical layers and the layers that undergo the largest weight changes.
- We propose IMM to mitigate this phenomenon. IMM offers a simple, general, and effective approach to integrate the reasoning improvements of the self-improved model while preserving the generalization of the original model.

2 Related Work

LLM Self-Improvement Given the high cost of labeling data, it is increasingly common to leverage LLMs to generate synthetic responses for training student models. Traditionally, this process has focused on knowledge distillation from stronger teacher models (Yuan et al., 2023; Wu et al., 2024). More recently, studies have demonstrated that distilling from weaker models—referred to as weak-to-strong knowledge distillation—can be more beneficial for LLMs compared to distilling from stronger models, given the same computational budget (Bansal et al., 2024). Another emerging direction is LLM self-improvement, where models improve themselves using their own outputs (Huang et al., 2022; Gulcehre et al., 2023; Singh et al., 2023). In the context of reasoning tasks, various self-improvement methods have been proposed: SPO (Prasad et al., 2024) employs Self-Consistency Preference Optimization for self-improvement; Pang et al. (2024) iteratively generate and refine data to optimize the model's reasoning ability; and Hosseini et al. (2024) utilize both correct and incorrect answers to improve reasoning performance through training an additional verifier.

Model Collapse As real-world data becomes increasingly scarce (Sutskever, 2024), synthetic data is playing a crucial role in training modern generative models due to its low cost and infinite availability. However, recent studies have revealed the risks associated with this "free lunch," a phenomenon known as model collapse (Shumailov et al., 2023). The model collapse has been extensively identified and analyzed in both computer vision (Hataya et al., 2023; He et al., 2022; Bohacek and Farid, 2023) and natural language processing (Alemohammad et al., 2024; Gerstgrasser et al., 2024). Researchers have investigated its underlying causes from both empirical (Padmakumar and He, 2024; Guo et al., 2023) and theoretical perspectives (Yuan et al., 2024; Bertrand et al., 2023; Seddik et al., 2024; Fu et al., 2024a). Current approaches to mitigating model collapse predominantly focus on data-centric methods. Feng et al. (2025) show that imperfect verifiers can help prevent model collapse by selecting appropriate data. Shumailov et al. (2023) proposes mixing data from previous iterations to prevent performance degradation, while Gerstgrasser et al. (2024) demonstrates that accumulating synthetic data over iterations reduces the risk of collapse.

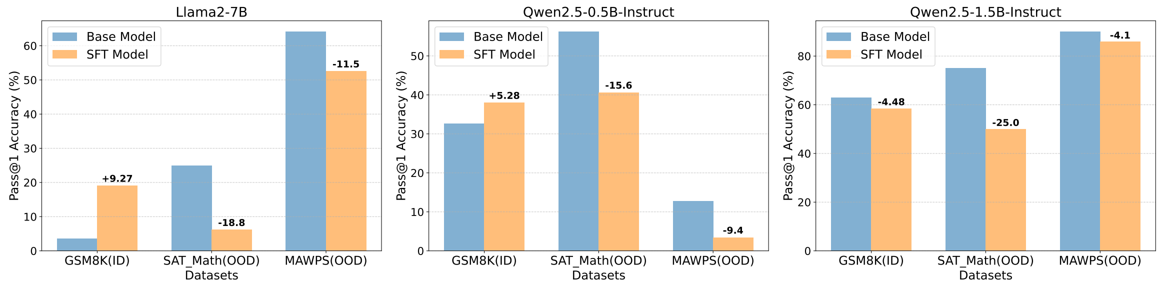

Figure 2: Superficial Self-improved Reasoners. The model's performance is only improved on in-domain reasoning datasets while losing the generalized reasoning capabilities on out-of-domain reasoning datasets.

Appendix C.9 discusses additional related works on LLM for reasoning. The connection with catastrophic forgetting is discussed in Appendix C.3.

3 Superficial Self-improved Reasoners

A natural and critical question arises for LLM self-improvement: does learning from synthetic reasoning data generated by the model itself trade off generalization ability for improved reasoning performance because of learning from itself? Our study shows that the answer is yes. In this section, we first confirm that self-improvement enhances in-domain reasoning performance but degrades general reasoning capabilities. We then investigate the underlying cause of this phenomenon by analyzing the layer-wise importance of the model during reasoning and tracking weight changes throughout the self-improvement process. A detailed comparison reveals a notable mismatch: the layers most crucial for reasoning experience relatively small weight updates, while less critical layers undergo more significant changes. This suggests that strong reasoning layers fail to substantially improve their reasoning ability through weight updates, whereas less important layers tend to overfit the training data rather than truly learning to reason.

3.1 Identify Superficial Self-improved Reasoners from OOD datasets

In this part, we identify Superficial Self-improved Reasoners by self-improving LLMs on the ID reasoning datasets and test them on OOD datasets.

Synthesizing Reasoning Data for Self-improvement We begin by establishing the self-improvement framework through the generation of reasoning data. Following prior work (Zelikman et al., 2022), we first synthesize reasoning data for fine-tuning. Let $\mathcal{D} = \{(q_i, a_i)\}_{i=1}^{n_d}$ denote a training dataset containing n_d reasoning questions q_i and corresponding final answers a_i. We also use Chain-of-Thought prompting (Wei et al., 2022) in this process (details in Appendix A.1). In the second step, we sample multiple solutions for each q_i using non-zero sampling temperatures, resulting in a synthetic dataset $\mathcal{D}_S = \{(q_i, \{(\hat{r}_{ij}, \hat{a}_{ij})\}_{j=1}^{k})\}$, where k represents the number of sampled solutions. Here, $\hat{r}_{ij}$ denotes the j-th reasoning path (i.e., rationale) generated by the model for q_i, and $\hat{a}_{ij}$ is the model's corresponding final answer. Incorrect solutions are then filtered out by comparing the sampled answers $\hat{a}_{ij}$ with the ground-truth answers a_i. Finally, we fine-tune the model on the filtered dataset $\tilde{\mathcal{D}}_G$ using supervised fine-tuning (SFT) to maximize the likelihood of generating reasoning paths r, optimizing the following objective:

$$\mathbb{E}_{(q,r,a)\sim\tilde{\mathcal{D}}_G}\left[\log p_\theta(r, a|q)\right]. \quad (1)$$

Loss of Generalized Reasoning Ability during Self-Improvement After applying the self-improvement framework to LLMs of various scales on ID datasets, we evaluate their performance on OOD reasoning datasets. The results, presented in Figure 2, reveal that while self-improvement enhances reasoning performance on ID datasets, it leads to a noticeable decline in performance on OOD datasets. This phenomenon suggests that although self-improvement improves metrics on ID reasoning tasks, it fails to enhance generalized reasoning capabilities and may even degrade them. We refer to this behavior as the emergence of *Superficial Self-Improved Reasoners*.

3.2 Investigating the Causes of Superficial Self-Improved Reasoners

While numerous studies on catastrophic forgetting focus on analyzing and addressing OOD perfor-

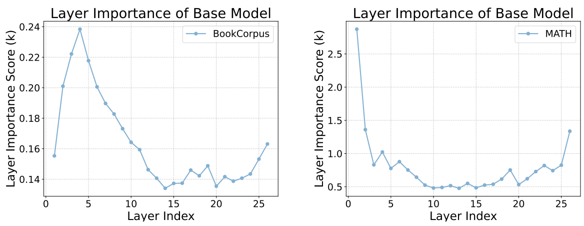

Figure 3: The Layer Importance Scores of strong reasoning model Qwen2.5-1.5B-Math on BookCorpus (left) and MATH datasets (right). The middle layers are less important while the early and late layers are more important for reasoning (MATH). For non-reasoning task (BookCorpus) middle layers are more important.

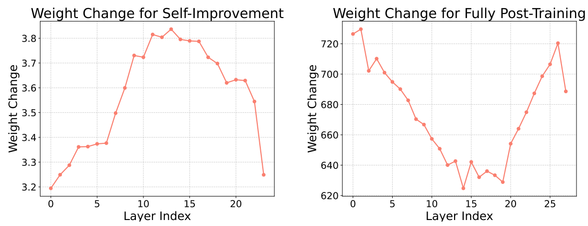

Figure 4: The weight change for SFT Qwen2.5-1.5B with self-improvement MATH data (left) and fully post-training Qwen2.5-1.5B to Qwen2.5-1.5B-Math using real data with 700B tokens (right).

mance degradation in continual learning for learning simpler tasks, our work specifically targets the more challenging domain of mathematical reasoning in LLMs, with an emphasis on understanding the phenomenon of Superficial Self-Improved Reasoners. In this section, we identify the most critical layers for reasoning, analyze how their weights evolve during the self-improvement process, and provide an explanation for the emergence of Superficial Self-Improved Reasoners.

Layer Importance for Reasoning To identify the most important weights in LLMs for reasoning, our objective is to determine and remove the weights that have the greatest impact on the model's prediction, which can be measured by the resulting change in loss. We denote the linear weight matrix as $\mathbf{W}^{k,n} = \left[W_{i,j}^{k,n}\right]$, where k represents the modules (e.g., a key projection in the multi-head attention (MHA) or an up-projection in the feed-forward network (FFN)) within the n-th LLM layer. We quantify the importance of each weight by measuring the error introduced when the corresponding parameter is removed. Given an in-domain reasoning dataset $\mathcal{D}$, the importance score $I_{i,j}^{k,n}$ for the weight $W_{i,j}^{k,n}$ is defined as:

$$\begin{aligned} I_{i,j}^{k,n} &= |\Delta\mathcal{L}(\mathcal{D})| \\ &= \left| \frac{\partial \mathcal{L}(\mathcal{D})}{\partial W_{i,j}^{k,n}} W_{i,j}^{k,n} - \frac{1}{2} W_{i,j}^{k,n} H_{kk} W_{i,j}^{k,n} \right. \\ &\quad \left. + \mathcal{O}\left(\|W_{i,j}^{k,n}\|^3\right) \right|. \end{aligned} \tag{2}$$

However, due to the significant computational cost associated with the large number of parameters in LLMs, we approximate the Hessian matrix H_{kk} using the Fisher information matrix, following the approach in Ma et al. (2023). This allows us to approximate the second-order term $\frac{1}{2} W_{i,j}^{k,n} H_{kk} W_{i,j}^{k,n}$ as $\frac{1}{2}\sum_{j=1}^{N}\left(\frac{\partial \mathcal{L}(\mathcal{D}_j)}{\partial W_i^k} W_i^k\right)^2$. By omitting the second-order derivative, the importance score $I_{i,j}^{k,n}$ is simplified to: $I_{i,j}^{k,n} \approx \left|\frac{\partial \mathcal{L}(\mathcal{D})}{\partial W_{i,j}^{k,n}} W_{i,j}^{k,n}\right|$. To assess the contribution of each layer to reasoning, we define the layer importance score as:

$$I^n = \sum_{W_{i,j}^{k,n}} \left| \frac{\partial \mathcal{L}(\mathcal{D})}{\partial W_{i,j}^{k,n}} W_{i,j}^{k,n} \right|. \tag{3}$$

We leverage this layer importance score I^n to identify which layers contribute most significantly to reasoning tasks. As illustrated in Figure 3, the middle layers are less important while the early and late layers are more important for the reasoning (MATH) tasks. We also find similar performance on code reasoning tasks, as illustrated in Appendix B.2. However, for the non-reasoning dataset BookCorpus, the middle layers are more important. This observation highlights the early and late layers as *reasoning-critical layers* (More clarification for this term is in Appendix C.4), distinguishing their specialized function in reasoning.

Layer Weight Change after Self-Improvement After fine-tuning the LLMs on reasoning data, the weights are updated, enabling the model to learn reasoning capabilities. We now analyze these weight changes. Let $\Delta\mathbf{W}^n$ represent the total weight change at the n-th layer after SFT:

$$\Delta\mathbf{W}^n = \sum_k \left\| \mathbf{W}^{k,n} - \mathbf{W}_{\text{SFT}}^{k,n} \right\|, \tag{4}$$

where $\mathbf{W}^{k,n}$ denotes the original k-th weight matrix and $\mathbf{W}_{\text{SFT}}^{k,n}$ is the fine-tuned weight matrix. Figure 4 illustrates the weight change $\Delta\mathbf{W}^n$ across different layers. For the self-improved model, the

Model	Reasoning-Critical Layer	Most Weight Change Layer	Generalized Reasoning Capability
Self-Improved	Early, late	Middle	✗
Fully Post-trained	Early, late	Early, late	✓

Table 1: Comparison of self-improved model and fully post-trained math model.

largest weight change occurs in the middle layers. In contrast, for the math model which is fully post-trained with stronger generalized reasoning capability, the most significant weight changes are concentrated in the early and late layers. A similar condition happens for real data with limited training data size, as analyzed in Appendix B.1.

Takeaway By analyzing Figure 3 and Figure 4 (left), we observe that the middle layers (reasoning-trivial layers) are the least important for the strong reasoning capabilities of LLMs, yet these layers undergo the most significant updates during the self-improvement process. This phenomenon highlights a contradiction in how reasoning ability is acquired. If the model were solely learning generalized reasoning, the most substantial weight updates would occur in the early and late layers (reasoning-critical layers), as observed in fully post-trained math models with strong generalized reasoning capabilities, rather than in the middle layers.

This observation suggests that during self-improvement, the model does not exclusively enhance its reasoning ability but also exhibits a tendency to overfit the training data, effectively "memorizing" it. This overfitting behavior explains the improved performance on ID datasets while compromising the model's generalization to OOD tasks. The performance comparison in Figure 2 further supports this conclusion. We summarize all experimental findings in Table 1, which leads to the following key insights: *(i)* during self-improvement on reasoning tasks, LLMs may show improved reasoning performance on ID tasks but lose generalized reasoning ability on OOD tasks; *(ii)* This phenomenon arises from a mismatch between the reasoning-critical layers and the layers with significant weight changes, suggesting that the model memorizes the training data rather than truly learning generalized reasoning capability. We further provide analysis on the reasons for this **mismatch phenomenon** in Appendix C.2.

4 Superficial Self-improved Reasoners Benefit from *Iterative Model Merging*

Iterative Model Merging (IMM) In this section, we propose Iterative Model Merging (IMM) to mitigate the *Superficial Self-Improved Reasoners* phenomenon, as illustrated in Figure 5. In the first self-improvement iteration, we self improve the original base model and merge the resulting SFT model $\boldsymbol{\theta}^{\mathbf{0}}_{\boldsymbol{SFT}}$ with the base model $\boldsymbol{\theta}$ to obtain the merged model $\boldsymbol{\theta}^{\mathbf{0}}_{\boldsymbol{m}}$. In each subsequent iteration t ($t > 0$), we continue the self-improvement process by fine-tuning the previously merged model $\boldsymbol{\theta}^{\boldsymbol{t-1}}_{\boldsymbol{m}}$. The resulting self-improved model $\boldsymbol{\theta}^{\boldsymbol{t}}_{\boldsymbol{SFT}}$ is then merged with the original base model to obtain the updated merged model $\boldsymbol{\theta}^{\boldsymbol{t}}_{\boldsymbol{m}}$. To formally describe this process, we define the parameter change $\boldsymbol{\delta}^{\boldsymbol{t}}$ during each SFT iteration as follows:

$$\boldsymbol{\delta}^{\boldsymbol{t}} = \begin{cases} \boldsymbol{\theta}^{\boldsymbol{t}}_{\boldsymbol{SFT}} - \boldsymbol{\theta}^{\boldsymbol{t-1}}_{\boldsymbol{m}}, & \text{if } t > 0, \text{SFT } \boldsymbol{\theta}^{\boldsymbol{t-1}}_{\boldsymbol{m}}, \\ \boldsymbol{\theta}^{\boldsymbol{t}}_{\boldsymbol{SFT}} - \boldsymbol{\theta}, & \text{if } t = 0, \text{SFT } \boldsymbol{\theta}. \end{cases} \tag{5}$$

We then incorporate DARE (Yu et al., 2024a) to further process $\boldsymbol{\delta}^{\boldsymbol{t}}$. DARE identifies parameter redundancy in LLMs, randomly masking parameter changes at a drop rate p while scaling the remaining updates to improve the performance of the merged model. Denoting $\mathbf{m} \sim \text{Bernoulli}(p)$, DARE can be expressed as:

$$\tilde{\boldsymbol{\delta}}^{\boldsymbol{t}} = (1 - \mathbf{m}) \odot \boldsymbol{\delta}^{\boldsymbol{t}}, \quad \hat{\boldsymbol{\delta}}^{\boldsymbol{t}} = \tilde{\boldsymbol{\delta}}^{\boldsymbol{t}} / (1 - p).$$

By incorporating DARE into our iterative model merging framework, the final update for each iteration t is given by:

$$\boldsymbol{\theta}^{\boldsymbol{t+1}}_{\boldsymbol{m}} = \alpha\boldsymbol{\theta} + (1 - \alpha)(\boldsymbol{\theta}^{\boldsymbol{t}} + \hat{\boldsymbol{\delta}}^{\boldsymbol{t}}), \tag{6}$$

where α is a scaling parameter that controls the balance between the base model weights and the self-improved model weights. Although we use a uniform α for all layers, which makes reasoning-critical layers's weight change remain minimal at the first iteration, this generalized way makes the model avoid overfitting and learn the generalized reasoning capability, which makes reasoning-critical layers' weights change increase more compared to the reasoning-trivial layers in the next iterations to learn generalized reasoning capability, as analyzed in Appendix B.9. The overall merging strategy is scalable for multiple iterations and larger models, with complexity analysis presented in Appendix B.10.

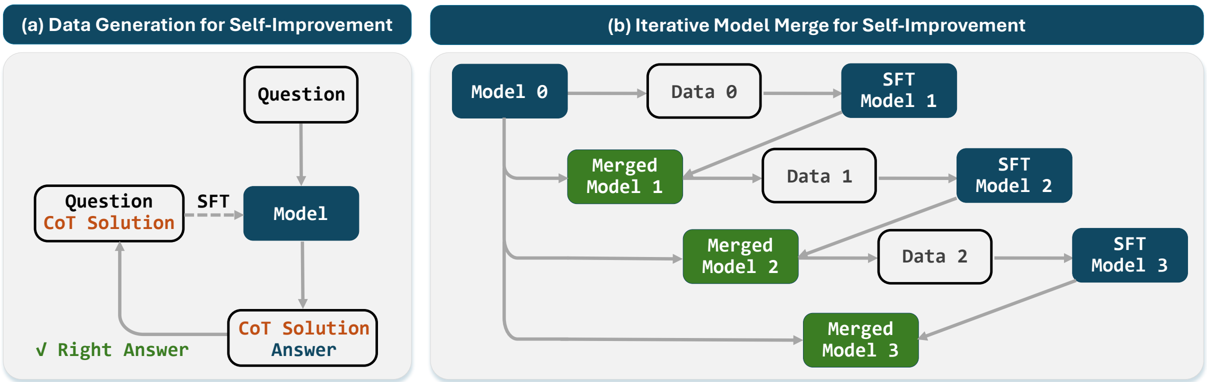

Figure 5: The overall framework: **(a)** The model generates chain-of-thought (CoT) answers for the given questions, and incorrect answers are filtered out using the ground-truth. The remaining correct answers are used for SFT to self-improve the model. **(b)** IMM iteratively SFT the model and merges the self-improved models with the base model to balance reasoning enhancement and generalization.

Insights for IMM The rationale behind model merging for generalized reasoning capability can be understood from two perspectives: *(i)* Based on the experimental observations in Section 3, the weights of reasoning-critical layers undergo significant changes during self-improvement, indicating that these layers are likely memorizing the training data. Given the blurred boundary between reasoning-critical and reasoning-trivial layers, it is plausible that middle layers also contribute to memorization, while late layers are partially involved in reasoning. As a result, excessive weight updates across all layers can lead to overfitting, especially when the training data is synthesized by the model itself. Model merging mitigates this overfitting by limiting weight changes. *(ii)* The base model retains strong generalization capabilities, while the self-improved model exhibits self-improved reasoning performance. Model merging combines the strengths of both, integrating the generalization ability of the base model with the reasoning improvements from the self-improved model.

Importance-based Iterative Model Merging (IIMM) We also propose IIMM, which is motivated to aggressively merge the model according to the layer importance as follows:

$$\boldsymbol{\theta}_{m,n}^{t+1} = \alpha\boldsymbol{\theta}_n + (1-\alpha)(\boldsymbol{\theta}_n^t + \frac{NI_n}{\sum_{n=1}^{N} I_n}\hat{\boldsymbol{\delta}}_n^t), \quad (7)$$

where n denotes n-th layer of the model with N layers. However, we find that IIMM is outperformed by IMM because of instability and overfitting datasets for importance score calculation. The detailed experiment and analysis are provided in Appendix B.5.

5 Experiments

In this section, we conduct extensive experiments to evaluate the effectiveness of our proposed method. Specifically, our experiments aim to address the following research questions: *(i)* Can our method prevent model collapse on complex reasoning tasks during iterative self-improvement? *(ii)* How well does our method perform on OOD reasoning tasks? *(iii)* Can our method be extended from self-improvement to knowledge distillation from a stronger model?

5.1 Setup

Datasets We train the model on MATH (Hendrycks et al., 2021) and GSM-8K (Cobbe et al., 2021) datasets correspondingly to evaluate the in-domain reasoning ability of the model, while evaluate it on MAWPS (Koncel-Kedziorski et al., 2016), SAT-Math (Zhong et al., 2024) datasets to evaluate the out-of-domain reasoning ability.

Models We include three LLMs at different scales (Qwen2.5-0.5B-Instruct, Qwen2.5-1.5B-Instruct (Yang and et al., 2024) and Llama2-7B (Touvron et al., 2023)) for self-improvement training. For the distillation experiments, we include stronger teacher models Qwen2.5-7B-Instruct for distillation. We also provide the recent model Llama3-8B performance in Appendix B.8

Baselines We evaluate our method by comparing it with four baselines. First, we consider **Vanilla** (STaR (Zelikman et al., 2022)) , which iteratively generates reasoning data following the procedure in Section 3 for self-improvement. Second, we include **Data Mixture** (Shumailov et al.,

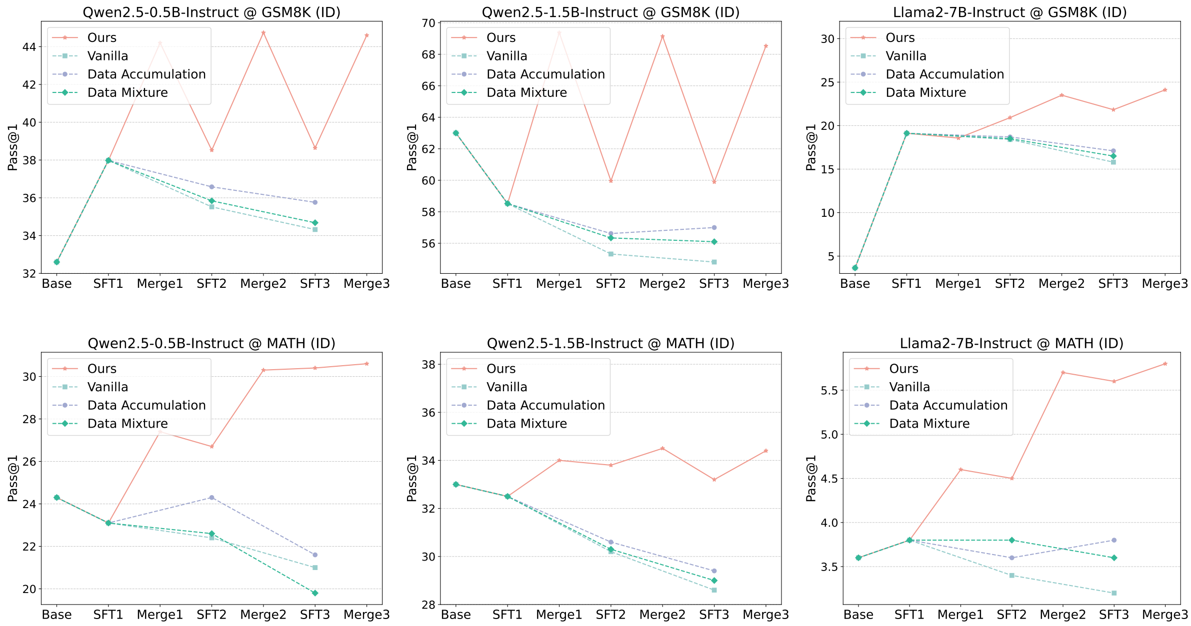

Figure 6: The model performances on in-domain (ID) datasets. SFT n and Merge n denote the SFT model and merged model in the n-th iteration cycle. The model collapse happens from the first or second iteration for baselines, while our method avoids it and achieves the best performance after model merging.

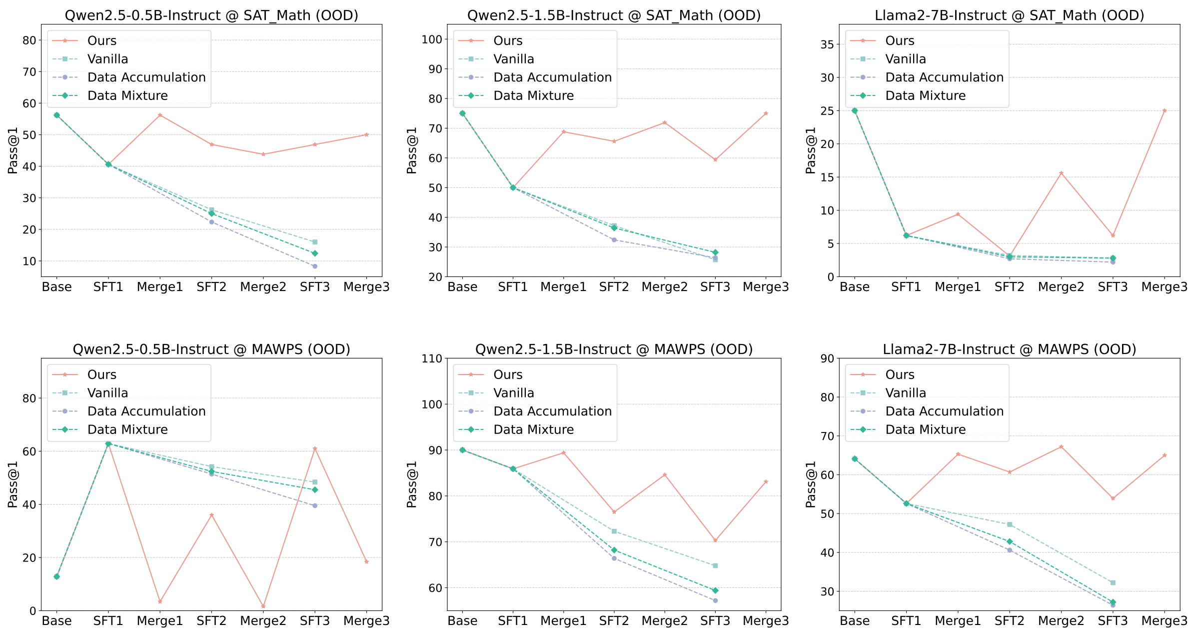

Figure 7: The model performances on out-of-domain (OOD) datasets. SFT n and Merge n denote the SFT model and merged model in the n-th iteration cycle. Baselines' performances decrease on most datasets, while IMM can generally maintain the OOD performance compared with the original base model.

2023), which mitigates performance degradation by mixing a portion of data from previous iterations. Third, we compare with **Data Accumulation** (Gerstgrasser et al., 2024), which demonstrates that accumulating synthetic data across iterations can prevent model collapse. We also provide a comparison of SFT interventions in Appendix B.4.

Evaluation We evaluate the model performance by computing pass@k $= \mathbb{E}_{\mathcal{D}_G}\left[1 - \frac{\binom{M-c}{k}}{\binom{M}{k}}\right]$, where c is the number of correct answers, out of total answer M and $\mathbb{E}_{\mathcal{D}_G}[\cdot]$ is the expectation for overall generated dataset $\mathcal{D}_G$. Therefore, pass@k measures the fraction of unique questions that have at least one correct answer when sampling k an-

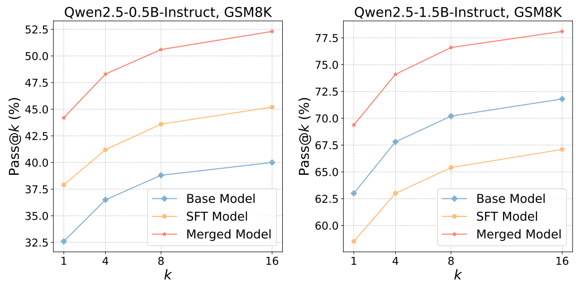

Figure 8: ID performance with different k for scaling up test-time-computing Pass@k on GSM8K.

Student	Domain	Datasets	Base	SFT	Merged
Qwen2.5-1.5B Instruct	ID	GSM8K	63.0	54.4	**71.6**
		MATH	24.3	**45.0**	42.6
	OOD	SAT_Math	75.0	75.0	**87.5**
		MAWPS	**90.0**	72.8	24.5
Llama2-7B	ID	GSM8K	3.6	**49.2**	38.8
		MATH	3.6	10.3	**12.5**
	OOD	SAT_Math	25.0	18.8	**28.1**
		MAWPS	64.1	55.1	**76.6**

Table 2: Student models' performance with distilling from stronger model setting. The best and runner-up accuracies are **bolded** and underlined respectively.

swers per question from the model.

Additional training and implementation details are provided in Appendix A.2.

5.2 ID Results with Self-improvement

To answer research question *(i)*, we conducted extensive experiments in a model collapse setting (iterative self-improvement) using two mathematical reasoning datasets, GSM8K and MATH. The results, shown in Figure 6, highlight that across three self-improvement iterations with three different LLMs, model collapse occurs in the first or second iteration for the baseline methods. In contrast, our method successfully avoids model collapse and achieves the best performance after applying model merging. Not only does our method significantly delay model collapse, but it also maintains superior performance across all iterations. Moreover, we observe that LLMs of all scales benefit from our model merging strategy, with smaller models suffering more severely from model collapse in the absence of this approach. Given the rising importance of test-time computing (Snell et al., 2025), we further evaluate our method by generating multiple answers and measuring pass@k accuracy. As shown in Figure 8 (more results are presented in Appendix B.7), our method consistently improves performance as k increases and outperforms both the base models and the SFT models.

5.3 OOD Generalization Results

To answer research question *(ii)*, We evaluate the checkpoints from Section 5.2 using OOD math reasoning datasets: SAT Math and MAWPS. Additional OOD datasets results can be found in Appendix B.3. The results, presented in Figure 7, show that while all other baselines suffer significant OOD performance degradation after iterative self-improvement, our method consistently restores performance after each model merging step and, in some cases, even surpasses the original base model.

The only exception is the Qwen2.5-0.5B-Instruct model on the MAWPS dataset. We hypothesize that this dataset closely resembles the in-domain data, where extensive ID training significantly improves performance, which causes a degradation during IMM. We further analyze this unexpected behavior in Appendix B.6. Overall, these results demonstrate the great potential of our method, as it successfully mitigates the generalization drop commonly observed during SFT.

5.4 Distillation from Stronger Models

Considering self-improvement may be only one of paradigms for LLM distillation, we extend our method to a broader field to answer research question *(iii)*. We distill a stronger Qwen2.5-7B-Instruct model into the weaker Qwen2.5-1.5B-Instruct and Llama-2-7B models. The results in Table 2 demonstrate that IMM consistently improves or maintains comparable performance on ID tasks, while often achieving significant improvements in OOD performance. This indicates that IMM only preserves task-specific performance but also enhances the model's generalized reasoning ability when distilling from the teacher model.

6 Conclusion

This study identifies that self-improved LLM reasoners still have the model collapse risk and lack generalized reasoning capability on OOD datasets. Our analysis reveals that the weight changes of layers doesn't match the layer importance. This mismatch suggests that instead of solely learning to reason, the model also memorizes the training data. To address this issue, we propose the Iterative Model Merge and extensive experiments demonstrate the effectiveness of our method: it not only mitigates model collapse but also make model have generalized reasoning capability.

Limitations

The proposed Iterative Model Merging (IMM) method currently employs a fixed-weight merging mechanism between the original and self-improved models. However, more advanced strategies, such as dynamic or layer-adaptive merging, could provide further improvements. Additionally, although IMM has proven to be effective in maintaining generalized reasoning capabilities, it doesn't investigate the strategy of mixing real and synthetic data appropriately, which could further enhance the trade-offs between reasoning improvement and generalization. We leave the exploration of advanced merging mechanisms and the optimal mixture ratio of real and synthetic data for future work.

7 Acknowledgement

This material is based upon work supported by the National Science Foundation under grant No. 2229876 and is supported in part by funds provided by the National Science Foundation, by the Department of Homeland Security, and by IBM. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation or its federal agency and industry partners.

References

Sina Alemohammad, Josue Casco-Rodriguez, Lorenzo Luzi, Ahmed Imtiaz Humayun, Hossein Babaei, Daniel LeJeune, Ali Siahkoohi, and Richard Baraniuk. 2024. Self-consuming generative models go MAD. In *International Conference on Learning Representations*.

Aida Amini, Saadia Gabriel, Peter Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. 2019. Mathqa: Towards interpretable math word problem solving with operation-based formalisms. *arXiv preprint arXiv:1905.13319*.

Hritik Bansal, Arian Hosseini, Rishabh Agarwal, Vinh Q Tran, and Mehran Kazemi. 2024. Smaller, weaker, yet better: Training llm reasoners via compute-optimal sampling. *arXiv preprint arXiv:2408.16737*.

Quentin Bertrand, Avishek Joey Bose, Alexandre Duplessis, Marco Jiralerspong, and Gauthier Gidel. 2023. On the stability of iterative retraining of generative models on their own data. *arXiv preprint arXiv:2310.00429*.

Quentin Bertrand, Joey Bose, Alexandre Duplessis, Marco Jiralerspong, and Gauthier Gidel. 2024. On the stability of iterative retraining of generative models on their own data. In *The Twelfth International Conference on Learning Representations*.

Matyas Bohacek and Hany Farid. 2023. Nepotistically trained generative-ai models collapse. *arXiv preprint arXiv:2311.12202*.

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

DeepMind. 2024a. Ai solves imo problems at a silver medal level. *URL https://deepmind.google/discover/blog/ai-solves-imo-problems-at-silver-medal-level/*.

DeepMind. 2024b. Alphazero: Shedding new light on chess, shogi, and go. *URL https://deepmind.google/discover/blog/alphazero-shedding-new-light-on-chess-shogi-and-go/*.

Xingjian Diao, Chunhui Zhang, Keyi Kong, Weiyi Wu, Chiyu Ma, Zhongyu Ouyang, Peijun Qing, Soroush Vosoughi, and Jiang Gui. 2025. Soundmind: Rl-incentivized logic reasoning for audio-language models. *arXiv preprint arXiv:2506.12935*.

Elvis Dohmatob, Yunzhen Feng, Arjun Subramonian, and Julia Kempe. 2025. Strong model collapse. In *The Thirteenth International Conference on Learning Representations*.

Yunzhen Feng, Elvis Dohmatob, Pu Yang, Francois Charton, and Julia Kempe. 2025. Beyond model collapse: Scaling up with synthesized data requires verification. In *The Thirteenth International Conference on Learning Representations*.

Shi Fu, Sen Zhang, Yingjie Wang, Xinmei Tian, and Dacheng Tao. 2024a. Towards theoretical understandings of self-consuming generative models. In *Forty-first International Conference on Machine Learning*.

Yonggan Fu, Zhongzhi Yu, Junwei Li, Jiayi Qian, Yongan Zhang, Xiangchi Yuan, Dachuan Shi, Roman Yakunin, and Yingyan Celine Lin. 2024b. Amoeballm: Constructing any-shape large language models for efficient and instant deployment. *Advances in Neural Information Processing Systems*, 37:78299–78319.

Matthias Gerstgrasser, Rylan Schaeffer, Apratim Dey, Rafael Rafailov, Tomasz Korbak, Henry Sleight, Rajashree Agrawal, John Hughes, Dhruv Bhandarkar Pai, Andrey Gromov, Dan Roberts, Diyi Yang, David L. Donoho, and Sanmi Koyejo. 2024. Is model collapse inevitable? breaking the curse of recursion by accumulating real and synthetic data. In *First Conference on Language Modeling*.

Nate Gillman, Michael Freeman, Daksh Aggarwal, Chia-Hong Hsu, Calvin Luo, Yonglong Tian, and Chen Sun. 2024. Self-correcting self-consuming loops for generative model training. In *International Conference on Machine Learning*.

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*.

Ruohao Guo, Wei Xu, and Alan Ritter. 2024a. Meta-tuning LLMs to leverage lexical knowledge for generalizable language style understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*.

Yanzhu Guo, Guokan Shang, Michalis Vazirgiannis, and Chloé Clavel. 2023. The curious decline of linguistic diversity: Training language models on synthetic text. *arXiv preprint arXiv:2311.09807*.

Yanzhu Guo, Guokan Shang, Michalis Vazirgiannis, and Chloé Clavel. 2024b. The curious decline of linguistic diversity: Training language models on synthetic text. In *Findings of the Association for Computational Linguistics: NAACL 2024*.

Ryuichiro Hataya, Han Bao, and Hiromi Arai. 2023. Will large-scale generative models corrupt future datasets? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Ruifei He, Shuyang Sun, Xin Yu, Chuhui Xue, Wenqing Zhang, Philip Torr, Song Bai, and Xiaojuan Qi. 2022. Is synthetic data from generative models ready for image recognition? *arXiv preprint arXiv:2210.07574*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

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. In *Neural Information Processing Systems Datasets and Benchmarks Track*.

Arian Hosseini, Xingdi Yuan, Nikolay Malkin, Aaron Courville, Alessandro Sordoni, and Rishabh Agarwal. 2024. V-star: Training verifiers for self-taught reasoners. *arXiv preprint arXiv:2402.06457*.

Lanxiang Hu, Qiyu Li, Anze Xie, Nan Jiang, Ion Stoica, Haojian Jin, and Hao Zhang. 2024. Gamearena: Evaluating llm reasoning through live computer games. *arXiv preprint arXiv:2412.06394*.

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 Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. 2024. Openai o1 system card. *arXiv preprint arXiv:2412.16720*.

Can Jin, Tong Che, Hongwu Peng, Yiyuan Li, Dimitris Metaxas, and Marco Pavone. 2024. Learning from teaching regularization: Generalizable correlations should be easy to imitate. *Advances in Neural Information Processing Systems*, 37:966–994.

Can Jin, Hongwu Peng, Qixin Zhang, Yujin Tang, Dimitris N Metaxas, and Tong Che. 2025. Two heads are better than one: Test-time scaling of multi-agent collaborative reasoning. *arXiv preprint arXiv:2504.09772*.

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

Rik Koncel-Kedziorski, Subhro Roy, Aida Amini, Nate Kushman, and Hannaneh Hajishirzi. 2016. MAWPS: A math word problem repository. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Yujia Li, David Choi, Junyoung Chung, Nate Kushman, Julian Schrittwieser, Rémi Leblond, Tom Eccles, James Keeling, Felix Gimeno, Agustin Dal Lago, et al. 2022. Competition-level code generation with alphacode. *Science*.

Zhaoyi Li, Gangwei Jiang, Hong Xie, Linqi Song, Defu Lian, and Ying Wei. 2024. Understanding and patching compositional reasoning in llms. *Findings of the Association for Computational Linguistics*.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. *arXiv preprint arXiv:2305.20050*.

Zheyuan Liu, Guangyao Dou, Xiangchi Yuan, Chunhui Zhang, Zhaoxuan Tan, and Meng Jiang. 2025. Modality-aware neuron pruning for unlearning in multimodal large language models. *The 63rd Annual Meeting of the Association for Computational Linguistics*.

Zimu Lu, Aojun Zhou, Houxing Ren, Ke Wang, Weikang Shi, Junting Pan, Mingjie Zhan, and Hongsheng Li. 2024. Mathgenie: Generating synthetic data with question back-translation for enhancing mathematical reasoning of llms. *arXiv preprint arXiv:2402.16352*.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. 2023. Llm-pruner: On the structural pruning of large language models. *Advances in neural information processing systems*, 36:21702–21720.

Amil Merchant, Elahe Rahimtoroghi, Ellie Pavlick, and Ian Tenney. 2020. What happens to BERT embeddings during fine-tuning? In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*.

Shen-yun Miao, Chao-Chun Liang, and Keh-Yih Su. 2020. A diverse corpus for evaluating and developing English math word problem solvers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Marius Mosbach, Anna Khokhlova, Michael A. Hedderich, and Dietrich Klakow. 2020. On the interplay between fine-tuning and sentence-level probing for linguistic knowledge in pre-trained transformers. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*.

OpenAI. 2025. Openai o3-mini. *URL https://openai.com/index/openai-o3-mini/*.

Vishakh Padmakumar and He He. 2024. Does writing with language models reduce content diversity? In *The Twelfth International Conference on Learning Representations*.

Richard Yuanzhe Pang, Weizhe Yuan, He He, Kyunghyun Cho, Sainbayar Sukhbaatar, and Jason E Weston. 2024. Iterative reasoning preference optimization. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are NLP models really able to solve simple math word problems? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Archiki Prasad, Weizhe Yuan, Richard Yuanzhe Pang, Jing Xu, Maryam Fazel-Zarandi, Mohit Bansal, Sainbayar Sukhbaatar, Jason Weston, and Jane Yu. 2024. Self-consistency preference optimization. *arXiv preprint arXiv:2411.04109*.

Zhenting Qi, Mingyuan MA, Jiahang Xu, Li Lyna Zhang, Fan Yang, and Mao Yang. 2025. Mutual reasoning makes smaller LLMs stronger problem-solver. In *The Thirteenth International Conference on Learning Representations*.

Peijun Qing, Chongyang Gao, Yefan Zhou, Xingjian Diao, Yaoqing Yang, and Soroush Vosoughi. 2024. AlphaLoRA: Assigning LoRA experts based on layer training quality. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. 2020. Zero: memory optimizations toward training trillion parameter models. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*.

Mohamed El Amine Seddik, Suei-Wen Chen, Soufiane Hayou, Pierre Youssef, and Merouane Debbah. 2024. How bad is training on synthetic data? a statistical analysis of language model collapse. *arXiv preprint arXiv:2404.05090*.

Dachuan Shi, Yonggan Fu, Xiangchi Yuan, Zhongzhi Yu, Haoran You, Sixu Li, Xin Dong, Jan Kautz, Pavlo Molchanov, and Yingyan Celine Lin. 2025. Lacache: Ladder-shaped kv caching for efficient long-context modeling of large language models. In *Forty-second International Conference on Machine Learning*.

Dachuan Shi, Chaofan Tao, Anyi Rao, Zhendong Yang, Chun Yuan, and Jiaqi Wang. 2023. Crossget: Cross-guided ensemble of tokens for accelerating vision-language transformers. *arXiv preprint arXiv:2305.17455*.

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

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. 2018. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144.

Avi Singh, John D Co-Reyes, Rishabh Agarwal, Ankesh Anand, Piyush Patil, Xavier Garcia, Peter J Liu, James Harrison, Jaehoon Lee, Kelvin Xu, et al. 2023. Beyond human data: Scaling self-training for problem-solving with language models. *arXiv preprint arXiv:2312.06585*.

Charlie Victor Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2025. Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning. In *The Thirteenth International Conference on Learning Representations*.

Ilya Sutskever. 2024. Sequence to sequence learning with neural networks: what a decade. *Keynote at NeurIPS*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti

Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Yue Wang, Hung Le, Akhilesh Deepak Gotmare, Nghi DQ Bui, Junnan Li, and Steven CH Hoi. 2023. Codet5+: Open code large language models for code understanding and generation. *arXiv preprint arXiv:2305.07922*.

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

Tianhao Wu, Weizhe Yuan, Olga Golovneva, Jing Xu, Yuandong Tian, Jiantao Jiao, Jason Weston, and Sainbayar Sukhbaatar. 2024. Meta-rewarding language models: Self-improving alignment with llm-as-a-meta-judge. *arXiv preprint arXiv:2407.19594*.

Weiyi Wu, Xinwen Xu, Chongyang Gao, Xingjian Diao, Siting Li, Lucas A Salas, and Jiang Gui. 2025. Assessing and mitigating medical knowledge drift and conflicts in large language models. *arXiv preprint arXiv:2505.07968*.

An Yang and et al. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

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

Zhifan Ye, Kejing Xia, Yonggan Fu, Xin Dong, Jihoon Hong, Xiangchi Yuan, Shizhe Diao, Jan Kautz, Pavlo Molchanov, and Yingyan Celine Lin. 2025. Longmamba: Enhancing mamba's long context capabilities via training-free receptive field enlargement. *The Thirteenth International Conference on Learning Representations*.

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. 2024a. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In *International Conference on Machine Learning*.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng YU, Zhengying Liu, Yu Zhang, James Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. 2024b. Metamath: Bootstrap your own mathematical questions for large language models. In *ICLR*.

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*.

Chunhui Zhang, Yiren Jian, Zhongyu Ouyang, and Soroush Vosoughi. 2024. Working memory identifies reasoning limits in language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*.

Chunhui Zhang, Zhongyu Ouyang, Kwonjoon Lee, Nakul Agarwal, Sean Dae Houlihan, Soroush Vosoughi, and Shao-Yuan Lo. 2025a. Overcoming multi-step complexity in multimodal theory-of-mind reasoning: A scalable bayesian planner. In *Forty-second International Conference on Machine Learning*.

Chunhui Zhang, Zhongyu Ouyang, Xiangchi Yuan, Soroush Vosoughi, et al. 2025b. Growing through experience: Scaling episodic grounding in language models. *The 63rd Annual Meeting of the Association for Computational Linguistics*.

Huaixiu Steven Zheng, Swaroop Mishra, Xinyun Chen, Heng-Tze Cheng, Ed H. Chi, Quoc V Le, and Denny Zhou. 2024. Take a step back: Evoking reasoning via abstraction in large language models. In *The Twelfth International Conference on Learning Representations*.

Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. 2024. AGIEval: A human-centric benchmark for evaluating foundation models. In *Findings of the Association for Computational Linguistics: NAACL 2024*.

Yichu Zhou and Vivek Srikumar. 2021. A closer look at how fine-tuning changes bert. *arXiv preprint arXiv:2106.14282*.

A Training and Implementation Details

A.1 Chain of Thought Prompting for Data Synthesis

We use chain-of-thought prompting (Wei et al., 2022) to generate answers. For MATH and GSM8K datasets, we both give 10 examples in the instructions for in-context generation. The prompting examples are given in Table 11 and Table 12. We generate 3 candidate answers for GSM8K and 6 candidate answers for MATH to have comparable numbers of right answers.

A.2 Training Details

We use NVIDIA RTX 8 $\times$ A6000 to train the model with DeepSpeed (Rajbhandari et al., 2020) distributed training framework. The number of training epoch is 3 and per device training batch size is 4. The gradient accumulation steps are set to 4 and the learning rate is 2e-5. The warm-up rate is 0.03. We use mixed precision training with bf16. We use DeepSpeed to distribute supervised fine-tuning model with ZeRO3, which partitions all three model states. We also use the vLLM library (Kwon et al., 2023) to generate synthetic reasoning data with sampling temperature $\{0.2, 0.4, 0.6\}$ to balance the diversity and accuracy of generated answers. Note that we use all models, data, and training tools solely for research purpose, which are consistent with their intended use.

The Model merging parameter in Section 3 is set to 0.5 to balance the base model and self-improved model. We use the setting in Section 5.4 to do the parameter analysis for α. Table 3 shows that $\alpha = 0.5$ can achieve a good balance between ID and OOD performance.

α	0.1	0.3	0.4	0.5	0.6	0.7	0.9
GSM8K	3.7	24.5	32.4	38.8	40.4	43.3	**48.3**
MATH	3.7	8.4	10.8	**12.5**	12.5	11.8	10.3
SAT_Math	25.3	27.4	27.7	**28.1**	26.7	24.8	18.8
MAWPS	64.4	69.0	**76.8**	76.6	69.3	64.3	57.2

Table 3: The parameter analysis for α.

B Additional Experiments and Analysis

B.1 Superficial Reasoning Finetuning Exists When Real Data Is Limited

We also find that even using real but limited data, Superficial Reasoning Synthetic Finetuning still exists. As Figure 9 shows, the middle layers change most compared with the early and late layers, while Figure 3 already shows that early and late layers are more important for reasoning. However, utilizing real data prevents the model from overfitting itself by using self-generated data. This is also verified by Figure 9: the model's reasoning layer (early and late layers) changed more (learn more reasoning capability) when training with real data, the reasoning-trivial layers (middle layers)'s weight change is close to middle layers when training with synthetic data.

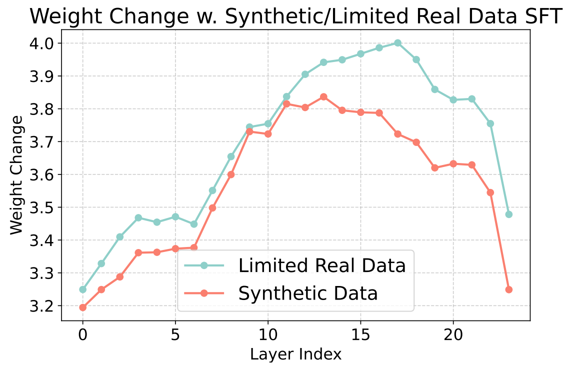

Figure 9: The weight change over layers for (i) Fintuning Qwen2.5-1.5B with synthetic MATH (Hendrycks et al., 2021) dataset data and limited training data (7.5k real MATH training data)

B.2 Layer Importance

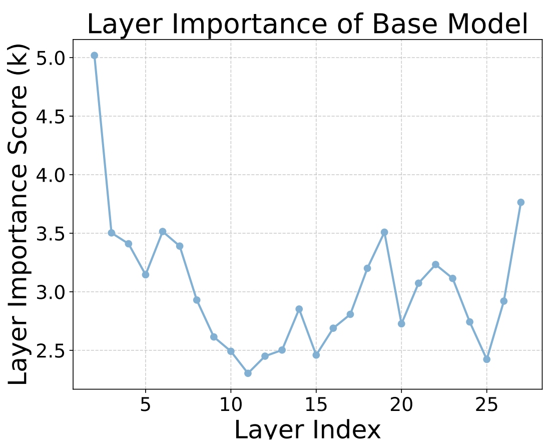

Figure 10: The layer importance score for Qwen2.5-1.5B base model on reasoning dataset MATH.

Here we provide the additional experiment results for evaluating the layer importance for Qwen2.5-1.5B base model on reasoning datasets MATH. Similar to stronger reasoning model Qwen2.5-1.5B-Math, the importance layer for reasoning is early and late layers, as demonstrated

Model	Datasets	Base	SFT1	Merge1	SFT2	Merge2	SFT3	Merge3
Qwen2.5-0.5B-I	SVAMP	7.3	35.8	5.9	21.6	1.3	**40.1**	9.8
	ASDiv	8.7	**51.4**	3.7	30.7	2.8	46.7	17.6
	MathQA	**37.9**	29.5	38.2	25.7	35.4	19.9	33.8
	MMLU_stem	34.2	34.6	38.4	34.3	**37.1**	27.9	36.1
Qwen2.5-1.5B-I	svamp	**77.7**	59	69.2	58.6	58.2	60.2	64.7
	asdiv	**82.8**	72.5	76.4	64.8	59.6	70.8	73.4
	MathQA	**62.5**	24.9	57.3	33.4	54.1	12.8	53.4
	MMLU_stem	53.6	40.1	52.6	47.9	53.4	41.7	**54.5**
Llama2-7B	svamp	39.6	30.1	38.0	35.1	39.0	33.5	38.5
	asdiv	51.9	42.9	51.2	46.7	52.3	41.4	**52.7**

Table 4: OOD performance on additional reasoning datasets.

Datasets	GSM8K	MATH	SAT_Math	MAWPS
Vanilla SFT	58.5	32.5	50.0	85.9
Gradient-decay (γ=0.9)	59.2	32.8	53.8	84.2
Gradient-clipping (max_norm=2.0)	58.7	31.7	52.3	84.7
Weight-masking (TopP=0.3)	60.2	**34.5**	56.2	87.0
IMM	**69.3**	34.0	**68.8**	**89.4**

Table 5: Qwen2.5-1.5B-Instruct performance compared with SFT interventions in the first iteration.

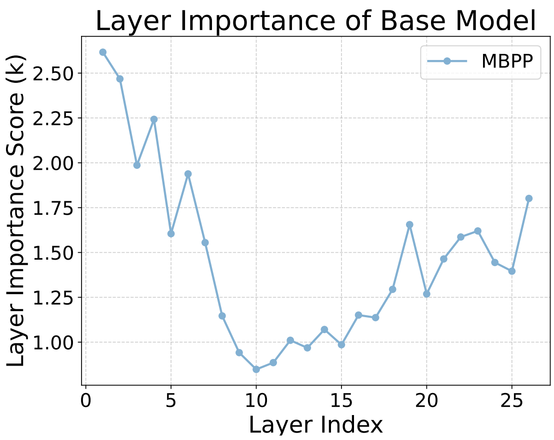

Figure 11: The layer importance score for Qwen2.5-1.5B base model on reasoning dataset MBPP.

in Figure 10. We also observed similar behavior in other complex reasoning task code generation MBPP, as demonstrated in Figure 11.

B.3 OOD Performance

We also provide OOD performance on additional datasets SVAMP (Patel et al., 2021), ASDiv (Miao et al., 2020), MathQA (Amini et al., 2019) and MMLU-stem Hendrycks et al. (2020). IMM keeps the OOD reasoning capability as shown in Table 4.

B.4 Comparison with SFT Interventions

We provide experimental results to compare these alternative interventions with IMM. As shown in Table 5, these methods generally do not outperform IMM, and in some cases are even outperformed by vanilla SFT. Compared with these interventions during SFT, IMM not only mitigate the overfitting reasoning finetuning, but also improve generalized reasoning capability through ensemble model merging. Our method is orthogonal to interventions for SFT, and provides a simple yet effective method to solve superficial self-improved reasoners phenomenon identified by this research.

B.5 Importance-based Weight Merge

Datasets	GSM8K	MATH	SAT_Math	MAWPS
I-IMM	**44.2**	**27.5**	49.3	2.1
IMM	**44.2**	27.4	**56.2**	**3.4**

Table 6: Comparison of IIMM and IMM across ID and OOD datasets.

We also experimented with weighting the merge ratio α per layer using the importance score I defined in Eq. (3). As shown in Table 6, this approach occasionally improves in-domain (ID)

Model	Base	SFT	Merge
Qwen2.5-0.5B	12.8	**32.9**	23.4
Qwen2.5-1.5B	**90.0**	72.8	24.5
Llama-2-7B	64.1	52.6	**65.3**

Table 7: Model performances on MAWPS dataset. The best performances are **bolded**, and the runner-up performances are underlined.

Dataset	Base	SFT	Merge
SAT_Math	75.0	75.0	**87.5**
MAWPS	**90.0**	72.8	24.5
MathQA	**62.5**	55.5	62.0
MMLU_stem	53.6	54.5	**57.6**
SVAMP	**77.7**	54.1	61.2

Table 8: Qwen2.5-1.5B-Instruct performance on external OOD datasets. The best performances are **bolded**, and the runner-up performances are underlined.

performance but often performs worse on out-of-domain (OOD) datasets. We hypothesize that this is because weighting the merging process based on ID-specific importance scores leads to overfitting to the ID data, thereby sacrificing the model's generalized reasoning capabilities on OOD tasks. Additionally, imbalanced merging rates across layers may introduce instability: when different layers are merged to varying degrees, the model can become internally inconsistent. In an extreme case, if some layers remain largely as base model layers while others are heavily adapted via SFT, this imbalance can degrade performance, as the layers are no longer "on the same page". We consider our merging method as a new way to increase the generalization of supervised learning, like other methods such as regularization (Jin et al., 2024) and meta-tuning (Guo et al., 2024a).

B.6 Analysis on Unexpected Behavior

OOD performance drops for Qwen2.5-1.5B on MAWPS dataset, and here we conduct more experiments to analyze this behavior. We found that (Table 7) small models (e.g., 0.5B and 1.5B) only suffer significant performance degradation on the MAWPS dataset after model merging. In contrast, larger models (e.g., 7B) achieve the best performance on MAWPS, benefiting more from IMM. Despite this drop on MAWPS, smaller models still show performance improvements on other OOD datasets. For instance, Table 8 shows that the 1.5B model outperforms both the Base and SFT versions on 5 OOD datasets. Therefore, we attribute the performance degradation on MAWPS primarily to two factors: (1) potential distributional differences in MAWPS compared to other datasets, and (2) the limited parameter capacity of small models, which may lack sufficient redundancy to support robust merging without trade-offs.

B.7 Additional Test-time Computing Results

We evaluate our method by generating multiple answers and measuring pass@k accuracy for MATH dataset. As shown in Figure 13, our method consistently improves performance as k increases and outperforms both the base models and the SFT models.

B.8 IMM with the Recent Model

Table 9 shows that for Llama3-8B model, IMM improves the ID performance and keeps comparable OOD performance, while vanilla SFT suffers from model collapse in ID datasets and severe degradation on OOD datasets.

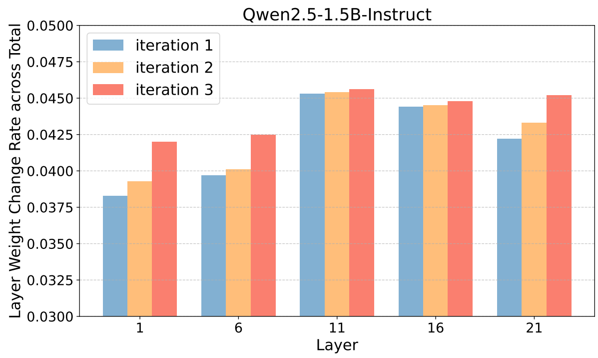

Figure 12: The percentage of the weight change over layers for finetuning Qwen2.5-1.5B in different iterations.

B.9 Weight Change for Different Iterations

After the first model weight merge, the parameter updates for the layers critical for reasoning still remain minimal, as illustrated in Figure 4. However, we continue to analyze the weight change across different layers and find that, although IMM uses an average merge rate across different layers, it improves the model's generalized reasoning capability, which makes the weight of reasoning-critical layers change more in the next iterations. Figure 12 shows that, in the next iterations, the reasoning-critical layers (early and late layers) change more weight change compared with the reasoning-trivial

Datasets	GSM8K	MATH	SAT_Math	MAWPS
Base	55.1	16.1	**53.1**	**90.8**
SFT	53.4	17.2	35.2	80.1
IMM	**61.2**	**19.5**	52.8	89.5

Table 9: Llama3-8B performance for the first self-improvement iteration. The best performances are **bolded**, and the runner-up performances are underlined.

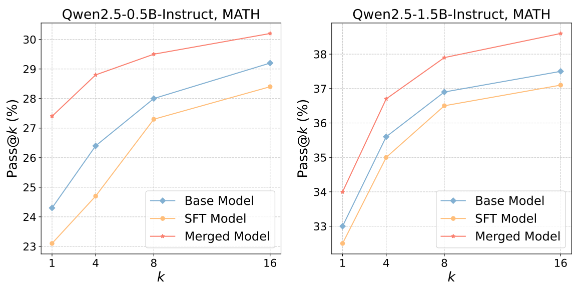

Figure 13: ID performance with different k for scaling up test-time-computing Pass@k on MATH.

layers (middle layers), indicating the model learns the generalized reasoning capability after IMM. Also, although IMM uses a uniform merge rate α across all layers, the absolute weight change difference between reasoning-critical layers and reasoning-trivial layers becomes smaller compared with SFT. This small difference accumulates over the course of the iterative self-improvement process. As a result, IMM achieves a relatively more balanced distribution of weight changes across layers compared to vanilla self-improvement and other baselines, where middle layers undergo disproportionately larger updates than early and late layers. IMM model therefore brings better generalized reasoning capability.

B.10 Complexity

Let n be the number of model parameters, T be the number of IMM iterations, $F(n)$ be the cost of one SFT training session. We calculate the complexity for IMM in the Table 10. The overall complexity is $\mathcal{O}(T \cdot F(n))$. Since fine-tuning dominates, especially for large models, the primary bottleneck is still the repeated SFT stages. Therefore, IMM introduce linear complexity on n, which can be overlooked compared with $\mathcal{O}(F(n))$, ensuring the scalability.

C Additional Discussion and Clarification

C.1 A Bitter Lesson: Not All LLMs Can Self-improve

During our experiment, we also find that not all the LLMs can self-improve on reasoning tasks. If LLM's performance decreases after SFT, then our method may not let the merged model have a better performance compared with the original model and the model after SFT. This usually happens when the original model already has a good performance (reasoning ability), and learned reasoning ability can't offset the generalization loss.

C.2 Why Importance-Weight Change Mismatch Happens?

We conclude two possible contributing factors to this observation: (i) Characteristics of SFT on Pre-trained LMs: Prior studies (Merchant et al., 2020; Mosbach et al., 2020; Zhou and Srikumar, 2021) have shown that during SFT, the early and late layers of pre-trained language models tend to undergo minimal changes. In particular, the late layers often preserve their original representations, suggesting a structural bias of SFT toward updating the middle layers. (ii) Inhibitory Effect of Self-improvement on Reasoning-critical Layers: As shown in Figure 9, models fine-tuned on real data exhibit more weight change in reasoning-critical layers (early and late layers) compared to those fine-tuned on self-synthesized data. In contrast, the middle layers show comparable levels of weight change in both settings. This indicates that the self-improvement process inherently inhibits updates to reasoning-critical layers, leading to disproportionate changes in the middle layers.

We further explain why middle layers contribute less to complex reasoning tasks. Prior work (Li et al., 2024) shows that weaker, implicit reasoning signals tend to surface in the middle layers, whereas stronger, explicit reasoning—such as chain-of-thought reasoning—emerges primarily in the late (and occasionally early) layers. In our study, to solve complex reasoning tasks model generated long CoT reasoning path, which depends on late layers

In summary, superficial self-improvement leads to overfitting on middle-layer representations where weaker, implicit reasoning resides, due to both the inherent bias of SFT and self-generated data. In contrast, reasoning-critical layers, responsible for explicit CoT reasoning, remain largely

Operation	Complexity
SFT	$\mathcal{O}(F(n))$
Compute δ^t	$\mathcal{O}(n)$
Masking, scaling	$\mathcal{O}(n)$
Merge update	$\mathcal{O}(n)$
Overall Complexity	$\mathcal{O}(T \cdot (F(n) + n)) \approx \mathcal{O}(T \cdot F(n))$

Table 10: Time complexity of IMM update steps.

unchanged, limiting the model's ability to improve on more complex reasoning tasks. Actually, it's very common that different layer has different behaviors for tasks (Qing et al., 2024), and our work emphasizes it in reasoning.

C.3 The Connection to Catastrophic Forgetting

Catastrophic forgetting is a related but distinct phenomenon compared to superficial self-improved reasoners. Specifically, catastrophic forgetting refers to the loss of previously acquired knowledge when deep learning models are trained on new data. This issue occurs because model parameters are optimized based on the most recent training data, causing earlier learned representations to be dramatically overwritten.

While both catastrophic forgetting and superficial self-improved reasoners result in degraded performance due to further fine-tuning, their effects differ. After fine-tuning on new data, catastrophic forgetting results in a performance loss on previously learned tasks, whereas superficial self-improved reasoners result in diminished generalization capabilities on out-of-domain (OOD) tasks. This discrepancy arises because in catastrophic forgetting, fine-tuning on data for new tasks causes the model to lose knowledge from previous tasks. In contrast, superficial self-improved reasoners do not lead to forgetting too much past information but instead shift towards overfitting due to potentially biased knowledge, which may self-enhance along with the iteration of synthesizing new data and fine-tuning on it.

C.4 The Definition for Layers

We do not provide a rigorous theoretical definition or external citation for the terms "reasoning-trivial layers" and "reasoning-trivial layers". In our paper, we adopt a relative and empirical definition: "reasoning-trivial layers" refer to the layers that exhibit lower importance scores in comparison to others, and "reasoning-trivial layers" refer to the layers that exhibit higher importance scores based on our layer-wise reasoning importance analysis. While not formally defined, this relative notion is sufficient for our purposes. It allows us to identify and analyze the mismatch between reasoning-critical layers (i.e., those with high importance scores) and the layers undergoing the most weight change during self-improvement. This mismatch is central to our discovery of the superficial self-improvement phenomenon.

C.5 Why This Importance Score

We would like to clarify that while the identification of key layers has been widely explored in prior work, such as in model analysis, pruning (Liu et al., 2025), and importance-based selection, our study does not aim to introduce a theoretical advancement in key layer selection itself. Rather, our contribution lies in uncovering a novel phenomenon: a mismatch between reasoning-critical layers and the layers experiencing the most weight change during self-improvement. We believe this observation offers a new perspective on how generalized reasoning capabilities may be hindered by superficial self-improvement. Building on this insight, we propose IMM as a method to mitigate this issue and improve the model's generalization in reasoning tasks.

Compared to other popular evaluation such as gradient change, the metrics defined in Eq. (3) and Eq. (4) are more suitable for the type of analysis conducted in this work. Specifically, Eq. (3) directly measures "how much the parameters of a given layer have actually changed from the beginning to the end of training." This provides a clearer indication of how much information is retained or adjusted through the self-improvement process, which is more aligned with our goal of understanding where learning occurs across the model. In contrast, gradient change is more appropriate for analyzing *how quickly* or *at which stage* the model learns during training. We appreciate the suggestion and agree that gradient analysis can provide complementary insights. We will include gradient tracking in manuscript to help monitor training stability and to identify potential issues such as exploding or vanishing gradients during self-improvement cycles.

C.6 SFT Overfitting

Overfitting is a common issue in supervised learning. Indeed, our work does not dispute or repeat this general principle. Rather, our contribution lies in highlighting a previously underexplored phenomenon: in particular, when applied to reasoning tasks, self-improvement via SFT tends to exacerbate generalization degradation more severely than standard SFT on curated or distilled data.

This distinction is central to our study. While overfitting is ubiquitous in supervised learning, our empirical results demonstrate that SFT using self-generated data amplifies this risk, leading to a sharper decline in reasoning generalization and more pronounced model collapse. This opinon is also supported by our interpretability analysis. To the best of our knowledge, this amplification effect of self-improvement has not been explicitly analyzed in prior work. To support this claim, we conducted systematic experiments beyond the self-improvement setting in Section 5.4 and Appendix B.1. We also demonstrate that even in general SFT settings like distillation, our proposed method still improves reasoning generalization, highlighting the broader applicability and potential of our approach.

C.7 Model Scale and Overfitting

While it is intuitive to assume that smaller models are more susceptible to overfitting, prior work (Dohmatob et al., 2025) suggests that overfitting severity depends non-monotonically on model size. Specifically, they report that models below a certain size threshold may exhibit pronounced overfitting, whereas models above that threshold may also overfit due to reduced margin for improvement on certain tasks.

In our study (Section 5), we scale model size to acknowledge this complexity and do not assert a universal trend. Our empirical results in Figure 6 and Figure 7 show inconsistent overfitting behavior across different model sizes and datasets, making it difficult to isolate model size as the primary factor behind model collapse. In fact, our cross-model and cross-dataset comparison suggests an interesting trend: when a model already performs strongly on a dataset, further supervised fine-tuning on self-generated (and correct) samples may inadvertently trigger overfitting due to reduced room for meaningful generalization.

C.8 How Other Methods Lead to Superficial Reasoning Learning

While existing approaches like data mixture (Shumailov et al., 2023), data accumulation (Gerstgrasser et al., 2024), adding real data (Dohmatob et al., 2025), and data selection (Guo et al., 2024b), have indeed been effective in expanding the distribution or sampling diversity, their application has primarily targeted non-reasoning tasks, for example, language modeling, open-ended generation, and summarization. These tasks are less sensitive to distributional sparsity and often benefit from simple accumulation or semantic filtering strategies that enhance diversity.

However, our work demonstrates that reasoning tasks are more vulnerable to model collapse due to complex reasoning tasks require structured, compositional, and often multi-hop inference capabilities. In such tasks, simply expanding diversity in a linguistic or syntactic manner often filters out complete and structured reasoning trajectories or repeat low quality data, with models failing to learn high-quality reasoning patterns with generalization. This is a critical gap that previous literature has not examined, especially in SFT reasoning data. This work diagnoses and mitigates model collapse for reasoning tasks under SFT, showing that naive distribution expansion can be ineffective or even worse. Our model interpretability analysis shows that using self-generated data will strengthen the harmful model weights updating. Therefore, data diversity method still hard to avoid it. In contrast, we propose that model merging provides a more robust and effective solution inspired by interpretability analysis. Empirically, we show that this strategy outperforms data-centric diversity enhancements in preserving general reasoning capability. This distinction between task types (reasoning vs. non-reasoning), model interpretability, and the limitations of prior methods is an important motivation of our work.

C.9 Related Works on LLM Reasoning

LLMs have demonstrated remarkable success across various reasoning tasks, including mathematical problem-solving, code generation, multimodality, agent, and common-sense reasoning (Yu et al., 2024b; Wang et al., 2023; Shi et al., 2023; Diao et al., 2025; Fu et al., 2024b; Wu et al., 2025; Shi et al., 2025; Jin et al., 2025; Ye et al., 2025; Diao et al., 2025; Zhang et al., 2025a,b). Beyond leveraging sophisticated prompting techniques to enhance reasoning capabilities (Kojima et al., 2022; Wei et al., 2022; Zheng et al., 2024; Yao et al., 2024), many methods focus on fine-tuning LLMs with reasoning datasets to create more robust reasoners (Lu et al., 2024; Yu et al., 2024b). For instance, approaches like SI (Huang et al., 2022), STaR (Zelikman et al., 2022), V-STaR (Hosseini et al., 2024), and rSTaR (Qi et al., 2025) fine-tune LLMs on task-specific datasets or synthesize reasoning data tailored for corresponding tasks. In addition to training models to generate correct answers, some studies introduce external verifiers (Cobbe et al., 2021; Lightman et al., 2023; Hosseini et al., 2024; Yuan et al., 2024) that select the best answer from a set of candidate solutions.

D Potential Risks

Enhancing LLMs with self-improving generalized reasoning capability may introduce risks of unintended capability emergence, including misuse in adversarial contexts such as misinformation or manipulation. As the model gains broader reasoning abilities across domains, it may be used for enabling harmful applications with enhanced reasoning capability. This highlights the importance of pairing IMM with safe evaluation and alignment to ensure safe and responsible deployment.

Prompt for Generating GSM8K Answers

Below is an instruction that describes a task.
Write a response that appropriately completes the request like given examples below:

Question: Angelo and Melanie want to plan how many hours over the next week they should study together for their test next week. They have 2 chapters of their textbook to study and 4 worksheets to memorize. They figure out that they should dedicate 3 hours to each chapter of their textbook and 1.5 hours for each worksheet. If they plan to study no more than 4 hours each day, how many days should they plan to study total over the next week if they take a 10-minute break every hour, include 3 10-minute snack breaks each day, and 30 minutes for lunch each day?

A: Let's think step by step.
Angelo and Melanie think they should dedicate 3 hours to each of the 2 chapters, 3 hours x 2 chapters = 6 hours total.
For the worksheets they plan to dedicate 1.5 hours for each worksheet, 1.5 hours x 4 worksheets = 6 hours total.
Angelo and Melanie need to start with planning 12 hours to study, at 4 hours a day, 12 / 4 = 3 days.
However, they need to include time for breaks and lunch. Every hour they want to include a 10-minute break, so 12 total hours x 10 minutes = 120 extra minutes for breaks.
They also want to include 3 10-minute snack breaks, 3 x 10 minutes = 30 minutes.
And they want to include 30 minutes for lunch each day, so 120 minutes for breaks + 30 minutes for snack breaks + 30 minutes for l unch = 180 minutes, or 180 / 60 minutes per hour = 3 extra hours.
So Angelo and Melanie want to plan 12 hours to study + 3 hours of breaks = 15 hours total.
They want to study no more than 4 hours each day, 15 hours / 4 hours each day = 3.75
They will need to plan to study 4 days to allow for all the time they need.
The answer is 4

Question: Mark's basketball team scores 25 2 pointers, 8 3 pointers and 10 free throws. Their opponents score double the 2 pointers but half the 3 pointers and free throws. What's the total number of points scored by both teams added together?
A: Let's think step by step.
Mark's team scores 25 2 pointers, meaning they scored 25*2= 50 points in 2 pointers.
His team also scores 6 3 pointers, meaning they scored 8*3= 24 points in 3 pointers
They scored 10 free throws, and free throws count as one point so they scored 10*1=10 points in free throws.
All together his team scored 50+24+10= 84 points
Mark's opponents scored double his team's number of 2 pointers, meaning they scored 50*2=100 points in 2 pointers.
His opponents scored half his team's number of 3 pointers, meaning they scored 24/2= 12 points in 3 pointers.
They also scored half Mark's team's points in free throws, meaning they scored 10/2=5 points in free throws.
All together Mark's opponents scored 100+12+5=117 points
The total score for the game is both team's scores added together, so it is 84+117=201 points
The answer is 201

Question: Bella has two times as many marbles as frisbees. She also has 20 more frisbees than deck cards. If she buys 2/5 times more of each item, what would be the total number of the items she will have if she currently has 60 marbles?
A: Let's think step by step.
When Bella buys 2/5 times more marbles, she'll have increased the number of marbles by 2/5*60 = 24
The total number of marbles she'll have is 60+24 = 84
If Bella currently has 60 marbles, and she has two times as many marbles as frisbees, she has 60/2 = 30 frisbees.
If Bella buys 2/5 times more frisbees, she'll have 2/5*30 = 12 more frisbees.
The total number of frisbees she'll have will increase to 30+12 = 42
Bella also has 20 more frisbees than deck cards, meaning she has 30-20 = 10 deck cards
If she buys 2/5 times more deck cards, she'll have 2/5*10 = 4 more deck cards.
The total number of deck cards she'll have is 10+4 = 14
Together, Bella will have a total of 14+42+84 = 140 items
The answer is 140

Other 5 examples here ...

Instruction:
Natalia sold clips to 48 of her friends in April, and then she sold half as many clips in May. How many clips did Natalia sell altogether in April and May?

Response: Let's think step by step.

Table 11: The CoT prompting examples for generate training data.

Prompt for Generating MATH Answers

Below is an instruction that describes a task.
Write a response that appropriately completes the request like given examples below:

Question: Kevin Kangaroo begins hopping on a number line at 0. He wants to get to 1, but he can hop only $\frac{1}{3}$ of the distance. Each hop tires him out so that he continues to hop $\frac{1}{3}$ of the remaining distance. How far has he hopped after five hops? Express your answer as a common fraction.
A: Let's think step by step.
Kevin hops $1/3$ of the remaining distance with every hop.
His first hop takes $1/3$ closer.
For his second hop, he has $2/3$ left to travel, so he hops forward $(2/3)(1/3)$.
For his third hop, he has $(2/3)^2$ left to travel, so he hops forward $(2/3)^2(1/3)$.
In general, Kevin hops forward $(2/3)^{k-1}(1/3)$ on his kth hop.
We want to find how far he has hopped after five hops.
This is a finite geometric series with first term $1/3$, common ratio $2/3$, and five terms.
Thus, Kevin has hopped $\frac{\frac{1}{3}\left(1-\left(\frac{2}{3}\right)^5\right)}{1-\frac{2}{3}} = \boxed{\frac{211}{243}}$.
The answer is \frac{211}{243}}

Question: What is the area of the region defined by the equation $x^2 + y^2 - 7 = 4y - 14x + 3$?
A: Let's think step by step.
We rewrite the equation as $x^2 + 14x + y^2 - 4y = 10$ and then complete the square,
resulting in $(x + 7)^2 - 49 + (y - 2)^2 - 4 = 10$,
or $(x + 7)^2 + (y - 2)^2 = 63$.
This is the equation of a circle with center $(-7, 2)$ and radius $\sqrt{63}$,
so the area of this region is $\pi r^2 = \boxed{63\pi}$.
The answer is 63\pi

Question: If $x^2 + y^2 = 1$, what is the largest possible value of $|x| + |y|$?
A: Let's think step by step.
If (x, y) lies on the circle,
so does $(x, -y)$, $(-x, -y)$, and $(-x, -y)$, (which all give the same value of $|x| + |y|$),
so we can assume that $x \geq 0$ and $y \geq 0$.
Then $|x| + |y| = x + y$. Squaring, we get
(x + y)^2 = x^2 + 2xy + y^2 = 1 + 2xy.
Note that $(x - y)^2 \geq 0$.
Expanding, we get $x^2 - 2xy + y^2 \geq 0$, so $2xy \leq x^2 + y^2 = 1$.
Hence,1 + 2xy \le 2,which means $x + y \leq \sqrt{2}$.
Equality occurs when $x = y = \frac{1}{\sqrt{2}}$,
so the maximum value of $|x| + |y|$ is $\boxed{\sqrt{2}}$.
The answer is \sqrt{2}

Other 5 examples...

Instruction:
If $f(x) = \frac{ax+b}{cx+d}, abcd \neq 0$ and $f(f(x)) = x$ for all x in the domain of f, what is the value of $a + d$?

Response: Let's think step by step.

Table 12: The CoT prompting examples for generating training data.