How to Mitigate Overfitting in Weak-to-strong Generalization?

Anonymous ACL submission

Abstract

Aligning powerful AI models on tasks that surpass human evaluation capabilities is the central problem of **superalignment**. To address this problem, weak-to-strong generalization aims to elicit the capabilities of strong models through weak supervisors and ensure that the behavior of strong models aligns with the intentions of weak supervisors without unsafe behaviors such as deception. Although weak-to-strong generalization exhibiting certain generalization capabilities, strong models exhibit significant overfitting in weak-to-strong generalization: Due to the strong fit ability of strong models, erroneous labels from weak supervisors may lead to overfitting in strong models. In addition, simply filtering out incorrect labels may lead to a degeneration in question quality, resulting in a weak generalization ability of strong models on hard questions. To mitigate overfitting in weak-to-strong generalization, we propose a two-stage framework that simultaneously improves the quality of supervision signals and the quality of input questions. Experimental results in three series of large language models and two mathematical benchmarks demonstrate that our framework significantly improves PGR compared to naive weak-to-strong generalization, even achieving up to 100% PGR on some models.

1 Introduction

Large language models (LLMs) have progressed rapidly in recent years, achieving superhuman ability in diverse tasks, and showing great potential in pursuing superhuman intelligence. Although large language models acquire extensive world knowledge and excellent capabilities to complete complex tasks through large-scale pre-training, alignment is still necessary to ensure that these models carry out tasks according to human intentions (Ouyang et al., 2022). The hard problem of alignment is “How do we align systems on tasks that are

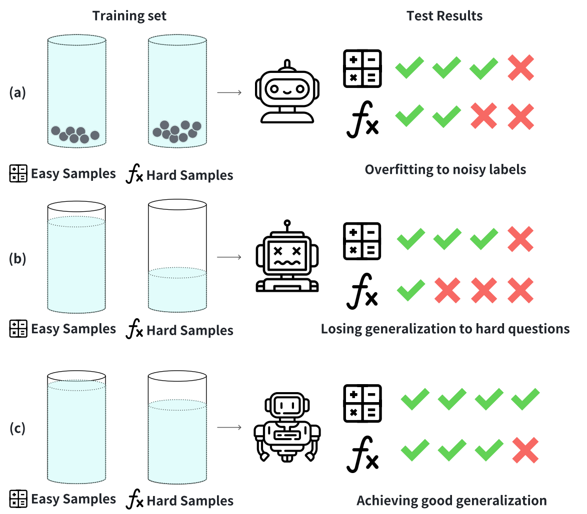


Figure 1: Illustration of different weak-to-strong generalization approaches. (a) Conventional approach with noisy labels from weak model, indicated by black dots; (b) Simple filtering approach that discards too many valuable hard samples; (c) Our framework can maintains both supervision quality and question quality.

difficult for humans to evaluate? (Leike, 2022) " This challenge is known as **superalignment**, which refers to how humans can align models on tasks that are beyond human ability to evaluate, which means that humans cannot provide correct supervision. One notable method in superalignment is the weak-to-strong generalization (Burns et al., 2023): **How can weak supervisors supervise stronger models?** This concept describes how the capacity of strong students can be elicited by fine-tuning on data labeled by weak teachers, consistently enabling them to outperform their weak teachers. In specific experiments, a weak model is typically used as a weak teacher, while a more capable model serves as the strong student.

Figure 1(a) demonstrates the features of weak-to-strong generalization, labels generated by the weak model contain noise due to its limited capabilities, thus presenting lower correctness and adding difficulties in eliciting strong model’s capabilities. As a

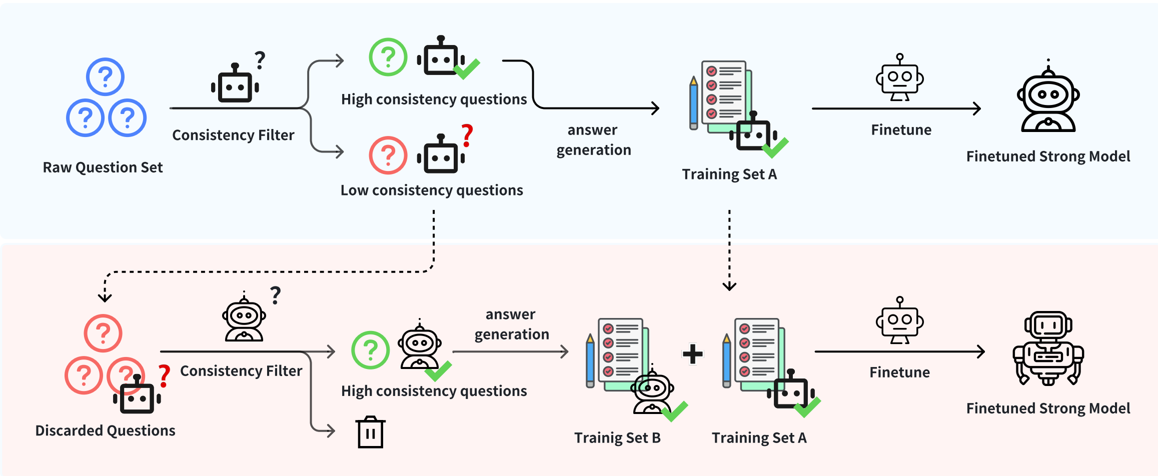


Figure 2: Overview of our two-stage training framework. **Stage I (top)**: The raw question set is filtered based on weak model's consistency (). High-consistency questions are used to generate Training Set A, which is then used for finetuning the strong model (). **Stage II (bottom)**: Previously discarded questions are re-evaluated and refined using the finetuned strong model from Stage I (). High-consistency questions are selected to form Training Set B, which is then combined with Set A for final finetuning (). Here represents weak model, represents primary strong model, represents Stage I finetuned model, and represents final finetuned model.

result, the strong model may overfit the erroneous weak supervisions, leading to performance degeneration (Yang et al., 2024a). Recent research has introduced filtering techniques to improve label correctness (Guo and Yang, 2024), making the analogy similar to easy-to-hard learning (Hase et al., 2024). In contrast to these related studies, we conduct a more in-depth investigation into the effects of commonly used data filtering methods. Based on our experimental results, we highlight that an excessive emphasis on data filtering can lead to data degeneration since some hard samples can be discarded, which may hinder the overall performance, as shown in Figure 1(b).In contrast, Figure 1(c) illustrates an ideal scenario, where a clean training set, containing both strong and weak samples, facilitates improved generalization. **These hard samples may be important to elicit student's capabilities to solve hard problems.**

For denoising supervision, most common methods, like filtering, tend to achieve better performance by improving supervision quality. However, such improvements come at the cost of lower question quality, harming features including difficulty and diversity, and overfiltering may even cause question degeneration.

Therefore, to mitigate overfitting and improve weak-to-strong generalization, we propose a two-stage weak-to-strong training framework, as depicted in Figure 2. In the first stage, we enhance supervision quality by filtering the generated samples based on weak model's uncertainty, which is estimated through the model's self-consistency. In the second stage, we further augment question quality by reusing the discarded questions and leverage the previous finetuned strong model to generate answers, as finetuned strong model may solve difficult questions better, incorporating those with high confidence back into the training dataset, to further elicit strong model's capabilities.

We assess the effectiveness of our framework on two popular mathematical reasoning benchmarks: GSM8k (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021). The evaluation involves two distinct model series: Llama 3 (Dubey et al., 2024) and Deepseek (Bi et al., 2024). The results demonstrate the substantial improvements offered by our framework. Specifically, the first stage outperforms the standard weak-to-strong method, while the second stage further enhances data quality and narrows the performance gap. On the commomly used criteria *performance gap recoverd (PGR)*, our framework significantly outperforms conventional weak-to-strong finetuning, reaching or surpassing 100% on certain models and datasets.

The main contributions of this paper are concluded as follows:

1. We pinpoint two critical factors for mitigat-

ing overfitting in weak-to-strong generalization: the quality of supervision and the quality of questions. And we demonstrate that enhancing supervision quality through data filtering leads to degeneration in question quality, which may harm the model's generalization on hard questions.

2. We introduce a two-stage weak-to-strong training framework focusing on supervision quality and question quality, effectively address overfitting on challenging reasoning tasks.

3. We conduct extensive experiments on MATH and GSM8k using model series including Llama 3 and Deepseek. The results demonstrate that our framework effectively mitigates overfitting, in which our first stage significantly outperforms the conventional weak-to-strong generalization method, and the second stage further enhances PGR with notable robustness, providing strong evidence of the effectiveness of our framework.

2 Background

In weak-to-strong generalization, the primary focus is how to elicit the ability of superhuman models using supervision from humans, as there is no access to superhuman tasks and superhuman models. The terms *Weak* and *Strong* here refer to model's latent potential, indicating human and superhuman models in the superalignment hypothesis.

Generally, the weak-to-strong generalization process involves the following steps, originally proposed by Burns et al. (2023):

1. Creating a weak supervisor: The weak supervisor referred to as *Weak Model*, is typically made by training small pretrained models. Its performance is referred to as *weak performance*.

2. Training strong models with weak labels: Data labelled by the weak model is used to finetune a large pretrained model, with the resulting performance termed *weak-to-strong performance*.

3. Training the strong ceiling: Ground truth data, used in the second step, is employed to finetune the large pretrained model, resulting in *strong ceiling performance*.

In the context of weak-to-strong generalization, the Performance Gap Recovered (PGR) is a commonly adopted criterion, introduced by Burns et al. (2023), to assess how effectively the potential of the strong model is elicited. A higher PGR indicates improved weak-to-strong performance, as it reflects the ability of the finetuned strong model to achieve performance closer to the "strong ceiling," thereby demonstrating the effective extraction of the model's full potential. The PGR is mathematically defined as:

$$PGR = \frac{\text{weak-to-strong} \; - \; \text{weak}}{\text{strong ceiling} \; - \; \text{weak}}. \quad (1)$$

In a specific model series, models' weak or strong can be directly represented by their model size, as a weak instruct model may outperform its strong under-elicited pretrained model, but still underperforms the strong finetuned model (e.g., Llama 3 8B Instruct vs Llama 3 70B & Llama 3 70B Instruct). In this work, we simplify weak supervisor's training by selecting the instruct versions of the current state-of-the-art models, as they show more human-like behaviours and generate more natural answers.

3 Methodology

An overview of our framework is illustrated in Figure 2. In the first stage, we use an uncertainty-based criterion to filter data labelled by the weak model, samples are filtered based on model's consistency and are then used to train the strong model. In the second stage, we reuse discarded questions showing high uncertainty for weak model in Stage I by employing the finetuned strong model to provide supervision. To ensure the correctness of the supervisions in Stage II, we also employ an uncertainty-based filtering criterion to retain the more accurate supervisory signals. Our framework simultaneously improves both the quality of supervision and the quality of questions in the weak-to-strong process, enhancing the generalization ability of weak-to-strong training.

3.1 Stage I: Purifying Supervision Signals

With given weak model M_{weak}, strong model M_{strong} and a set of questions, conventional weak-to-strong generalization directly use weak model to generate answers, then use generated samples to train strong model.However, due to weak model's limited ability, generated labels may contain many

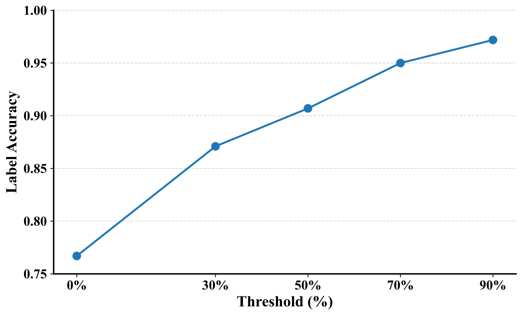


Figure 3: The relationship between supervision correctness and filtering threshold. As the filtering threshold increases, the supervision correctness (measured by label accuracy) shows a consistent upward trend.

noisy labels showing low supervision quality, causing overfitting during strong model finetune. To purify noisy supervision, we introduce an uncertainty-based filter, choosing samples with high model consistency. We employ chain-of-thought prompting to randomly generate ten responses for each question, thereby ensuring a diverse set of possible answers. Among these, we select the answer with the highest consistency as the model's final response, as it reflects the greatest confidence in the reasoning process. Specifically, for a selected answer Ans, which appears N_{Ans} times out of a total of N_{Total} samplings, the model's confidence in that answer is defined as:

$$\text{Confidence}(\text{Ans}) = \frac{N_{Ans}}{N_{Total}} \times 100\%. \quad (2)$$

To filter out noisy labels and improve supervision quality, we apply an uncertainty-based filter based on model‘s confidence. By filtering samples with a consistency threshold, we form a filtered dataset of high-confidence question-answer pairs, shown as "Training set A" in Figure 2, showing higher supervision quality. Our experiments show that with higher consistency threshold results in higher sample correctness, as shown in Figure 3. We finally use the filtered dataset to finetune strong model, expecting to solve the problem of overfitting on wrong labels.

We further analyzed the effectiveness of chain-of-thought prompting, detailed in Appendix C.1.

3.2 Stage II: Mitigating Question Degeneration

Following Stage I, the finetuned model $M_{\text{finretune}}$ and two distinct datasets are produced: a filtered dataset D_{filtered} containing high-certainty questions and a discarded dataset $D_{\text{discarded}}$ comprising low-certainty questions. The discarded questions often represent questions with higher difficulty or less common topics, where the weak model struggled to provide confident answers. Despite this, these questions remain crucial for improving overall model performance, as the test set typically encompasses a diverse range of difficulty levels and topics. Meanwhile, the finetuned model in Stage I, having its ability elicited by labels from weak teacher, now outperforms its weak teacher, showing the potential to solve questions beyond weak model's ability.

To address this, the finetuned student model—now exceeding the weak model in performance—is employed to generate answers for the discarded questions. For each question in the discarded question set, the finetuned model generates a variety of potential answers, providing a more accurate and comprehensive set of responses than its teacher. Similar to Stage I, an uncertainty-based filtering process is applied to retain only high-confidence samples, producing a high quality dataset, shown as "Training set B" in Figure 2.

The refined, high-certainty samples are then appended to the training set, creating an enriched dataset. This updated training set is subsequently used to finetune the initial strong model, enhancing its ability to generalize across the full spectrum of question difficulty and diversity. This refinement process ensures the inclusion of valuable but initially uncertain data, maximizing the strong model's potential and overall performance.

4 Experiments

4.1 Experimental Settings

Dataset We conduct experiments on two prominent mathematical reasoning benchmarks, the grade-school level reasoning task GSM8K (Cobbe et al., 2021) and the more challenging MATH task (Hendrycks et al., 2021). For training, we use the same training set as Yang et al. (2024b) for both weak model labelling and strong model training. For evaluation, we utilized the GSM8K evaluation set, which contains 1,319 data points. For MATH, we used the smaller subset as the primary evaluation test set following Lightman et al. (2024), which contains 500 data points. We compared the model's performance on the 500 samples subset with that on the original test dataset, with details provided in Appendix C.2.

Models We use several models to investigate the effectiveness of our framework, including the

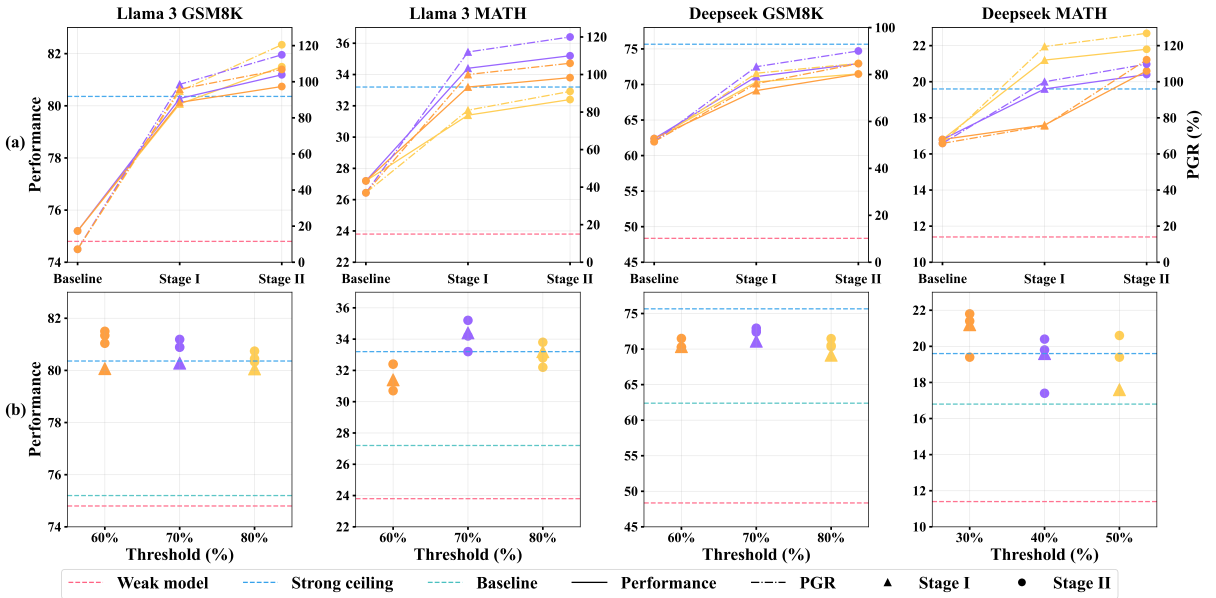


Figure 4: **(a)** The upper row shows the performance trajectory and PGR across different stages (Baseline, Stage I, and Stage II). The solid lines represent model performance (left y-axis), while the dash-dotted lines show PGR values (right y-axis). **(b)** The lower row demonstrates the impact of different filtering thresholds on model performance, with triangles representing Stage I results and circles representing Stage II results. For each experimental setting, points with the same color correspond to the same Stage I filtering threshold. Results show consistent improvement patterns across all model configurations, with Stage II generally achieving better performance than Stage I.

Llama 3 series (Dubey et al., 2024) (Llama 3 8B Instruct, Llama 3 70B) and the Deepseek series (Bi et al., 2024) (Deepseek 7B Chat, Deepseek 67B Base).

Evaluation Metrics We use accuracy and performance gap recovered (PGR) as our primary evaluation metrics. For PGR, we define the performance of small instruct/chat models as "weak performance", and the performance of strong models after finetuned with golden labels as "strong ceiling", each representing the starting and the goal performance we aim to achieve. Both metrics were employed to assess the effectiveness of the weak-to-strong generalization approach, highlighting the elicited abilities of the model and the extent to which the performance gap was recovered.

4.2 Main results

As illustrated in Figure 4, our framework significantly narrows the performance gap between finetuned strong model and strong ceiling, meanwhile effectively eliciting strong model's ability. Our experimental results demonstrate the efficacy of our framework across multiple model series, including Llama 3 and Deepseek. For the Llama 3 model, specifically the 70B variant, the performance in weak-to-strong generalization (PGR) on the GSM8K dataset shows a remarkable improvement, rising from 7.19% to 120.50% when utilizing the smaller Llama 3 8B Instruct model as the weak model. This improvement is accompanied by an increase in task performance, which climbs from 75.20% to 81.50%. Similar enhancements are observed on the MATH dataset, where PGR increases from 36.17% to 121.28% and task performance rises from 18.2% to 35.2%.

Comparable gains are seen with the Deepseek model series. On the GSM8K dataset, PGR increases significantly from 51.39% to 90.04%, while task performance improves from 62.39% to 72.94%. For the MATH dataset, PGR improves from 65.85% to 126.83%, with performance rising from 16.8% to 21.8%.

4.3 Performance Gains from Enhanced Supervision Quality

As illustrated in Figure 4(a), the uncertainty-based filtering approach implemented in Stage I consistently outperforms the conventional baseline across multiple datasets and model configurations. Specifically, for Llama 3 on the GSM8K dataset, the weak-to-strong generalization performance improves substantially from 7.19% to 98.56% in PGR, accompanied by an increase in task performance from 75.20% to 80.28%. On the MATH dataset, PGR rises from 36.17% to 112.77%, while task performance increases from 18.2% to 34.0%. Similarly, for Deepseek on GSM8K, PGR increases from 51.39% to 83.33%, while performance enhances from 62.39% to 71.11%. On the MATH dataset, Deepseek shows a notable improvement, with PGR rising from 65.85% to 119.51%, and task performance increasing from 16.8% to 21.2%.

4.4 Further Improvement from Enhanced Question Quality

As further illustrated in Figure 4(b), the refinement process in Stage II effectively enhances the quality of the training data, particularly in terms of difficulty and diversity, leading to significant improvements in model performance. Specifically, for the Llama 3 series, the strong model achieves a peak PGR of 120.50% on the GSM8K dataset, reflecting an additional 21.94% improvement compared to the finetuned strong model in Stage I, corresponding to a performance of 81.50%. On the MATH dataset, we observe a peak PGR of 121.28%, with a further increase of 8.51% compared to Stage I, reaching 35.2% on task performance.

For the Deepseek series, the strong model attains a peak PGR of 90.04% on GSM8K, marking an additional 6.71% improvement over Stage I, with a corresponding finetuned performance of 72.94%. On MATH, the peak PGR reaches 126.83%, demonstrating a further increase of 7.32% compared to Stage I, with task performance reaching 21.8%.

5 Analysis

5.1 The Impact of Excessive Filtering on Supervision Quality

As shown in Figure 3, label correctness increases as model uncertainty decreases. However, in preliminary experiments during Stage I, we observed an intriguing trend: while performance improves initially as uncertainty decreases, it starts to deteriorate after a certain threshold. This suggests that other factors, beyond supervision quality, influence weak-to-strong generalization, and existing filtering methods may have inherent limitations.

Reduction in Data Difficulty Figure 5 shows that increasing the filtering threshold leads to a decrease in average difficulty, with fewer hard questions (Levels 4-5) remaining in the dataset. These harder questions represent areas where the weak model is less confident, suggesting they are beyond its current capabilities. In contrast, easier questions (Levels 1-2), where the model is more confident, dominate the dataset. This results in a less challenging training set, hindering the model's ability to generalize to more difficult problems and contributing to data degeneration.

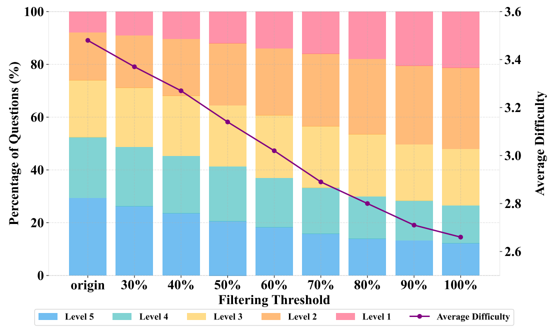


Figure 5: Impact of filtering threshold on question difficulty distribution. As the threshold increases, the proportion of difficult questions (Levels 4-5) decreases, while easier questions (Levels 1-2) increase, resulting in a decline in average difficulty from 3.48 to 2.66.

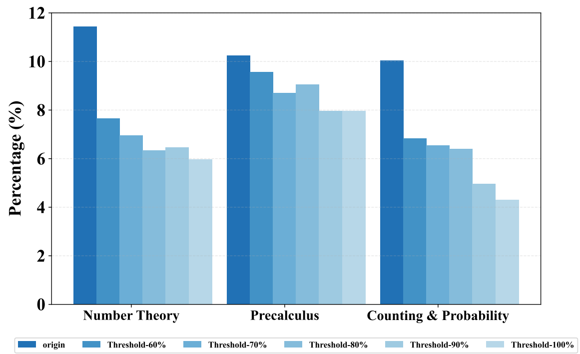
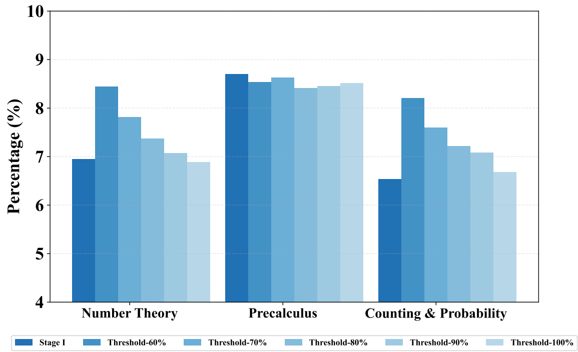


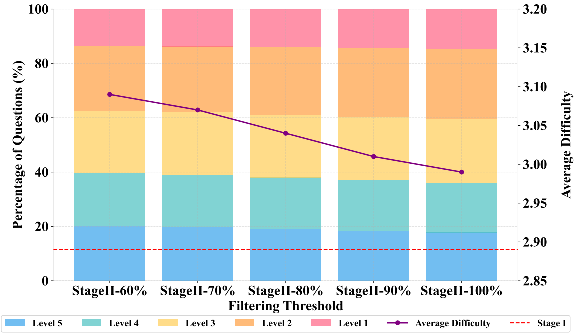
Figure 6: Changes in topic distribution across filtering thresholds for three representative mathematical categories. Filtering causes shifts in topic distribution, with minor categories seeing more reductions.

Shift in Data Diversity As shown in Figure 6, filtering also causes a significant shift in the diversity of questions. For instance, the Counting and Probability section drops from 10.79% to 4.31%, reflecting changes in the model's uncertainty. This shift in data diversity impacts the variety of question types, reducing exposure to harder topics. The complete trends and numerical results across all categories are provided in Appendix D.1.

Once the filtering threshold surpasses a certain point, performance degrades due to the exclusion of important, challenging data. While reducing label uncertainty can improve performance, excessive filtering diminishes the dataset's diversity, partic-



(a) Topic distribution comparison in Stage II under different thresholds.



(b) Distribution of difficulty levels and average difficulty scores in Stage II.

Figure 7: Difficulty and diversity analysis in Stage II (GSM8K, Llama 3, Threshold-70%), showing improved preservation of question quality.

ularly regarding difficulty and topic variety. This limits the model's ability to generalize effectively, leading to degeneration in its overall performance.

5.2 The Robust Effectiveness of Data Refinement in Stage II

To address excessive filtering, we propose a strategy that balances uncertainty-based filtering with the preservation of question quality, including difficulty and diversity. In Stage II, we regenerate answers for discarded questions from Stage I using the finetuned model, filtering them by uncertainty before adding low-uncertainty samples to the dataset.

As shown in Figure 4(a), Stage II consistently improves performance across all filtering thresholds, demonstrating the effectiveness of our framework in recovering lost data and boosting performance.

Figure 7 shows recovery in both difficulty and diversity, with the refined dataset closely resembling the original. For Llama 3 on MATH, PGR increases from 112.77% to 121.28%, and performance rises from 34.4% to 35.2%. Similar results are observed in Figure 4, highlighting the framework's robustness across models and datasets.

Additionally, Figure 4 demonstrates that even models with initially lower performance show significant improvements. For the Deepseek series on MATH, the performance gap between thresholds narrows in Stage II, indicating that the framework effectively recovers discarded data from over-filtered scenarios while refining fewer under-filtered questions.

5.3 The Importance of Label Filtering in Stage II

In Stage II, we focus on enhancing question quality and mitigating degeneration by using the finetuned model to generate answers for discarded questions from Stage I. Instead of adding all generated answers back, we apply an uncertainty-based filter to ensure only reliable answers are reintegrated, preventing the inclusion of low-quality data.

Table 1 summarizes the results of the ablation study comparing the framework with and without the filtering process, using the Llama 3 model series on the GSM8K dataset.

	Origin	With Filter	Without Filter
Stage I-50%	78.99	80.89 (+1.90)	78.31 (-0.68)
Stage I-60%	80.07	81.50 (+1.43)	78.84 (-1.23)
Stage I-70%	80.28	81.19 (+0.91)	80.28 (+0.00)
Stage I-80%	80.06	80.74 (+0.68)	79.59 (-0.47)

Table 1: The impact of **With** vs. **Without** label filtering in Stage II on Weak-to-Strong Generalization.

As seen in Table 1, appending all generated samples without filtering leads to performance degradation, highlighting that indiscriminate inclusion reduces supervision quality. The uncertainty-based filter ensures optimal supervision and question quality, which are critical for effective weak-to-strong reasoning generalization.

5.4 Exploring the Potential for Further Iterative Refinement

While our current framework is effective, we believe additional iterative refinement could further improve question quality and enhance overall performance. Specifically, the Stage II refinement process—where discarded questions are recovered and answered by the finetuned strong model—presents significant potential for iterative enhancement. As the model improves, this iterative process could continuously boost question quality.

We introduce an additional iteration, Stage Exp, where discarded questions are reprocessed using the finetuned model in Stage II to generate answers,

	Accuracy	PGR
GSM8K		
Baseline	62.39	51.39%
Stage I	71.11	83.33% (+31.94%)
Stage II	72.94	90.04% (+38.65%)
Stage Exp-Threshold-80%	72.26	87.55%
Stage Exp-Threshold-90%	72.93	90.00%
Stage Exp-Threshold-100%	**73.77**	**93.08% (+41.69%)**
MATH		
Baseline	16.8	65.85%
Stage I	21.2	119.51% (+53.66%)
Stage II	21.8	126.83% (+60.98%)
Stage Exp-Threshold-50%	21.4	120.71%
Stage Exp-Threshold-40%	21.2	119.51%
Stage Exp-Threshold-30%	**22.4**	**134.15% (+68.3%)**

Table 2: Performance comparison of iterative refinement on GSM8K and MATH datasets (Deepseek model). Best results are underlined.

followed by uncertainty filtering before reintegration into the dataset. Due to computational constraints, Stage Exp experiments were conducted on Deepseek series, focusing on best-performing configurations for GSM8K and MATH datasets.

As shown in Table 2, iterative refinement demonstrates a promising potential for further enhancement. However, determining the optimal threshold for these iterations remains an open challenge, which we plan to explore in future work.

6 Related Work

6.1 AI Deceptions

A persistent challenge in weak-to-strong generalization is AI deception, where strong models overfit to noisy labels from weak models, hindering their ability to generalize to complex samples (Yang et al., 2024a). A similar issue in reinforcement learning from human feedback (RLHF) is identified by Wen et al. (2024), where models mislead human evaluators.

This behaviour is akin to model sycophancy, where models align with human feedback at the expense of accuracy. Early work by Cotra (2021) and Perez et al. (2023) shows models often aim to please users. Sharma et al. (2024) attributes this to human preference biases. Solutions such as synthetic data (Wei et al., 2023) and pinpoint tuning (Chen et al., 2024) aim to mitigate sycophancy, while Sicilia et al. (2024) links it to model uncertainty.

6.2 Weak-to-Strong Generalization

Weak-to-strong generalization, introduced by OpenAI (Burns et al., 2023), has led to advancements in model training and supervision. Recent studies explore ensemble learning to improve labels by integrating predictions from smaller models (Liu and Alahi, 2024; Agrawal et al., 2024; Cui et al., 2024). Dong et al. (2024) enhances learning by replacing sample-label pairs with concept vectors, while Guo and Yang (2024) employs filtering and confidence-based reweighting. Additionally, Yang et al. (2024b) introduces a two-stage framework for refining training data, and Lyu et al. (2024) proposes multi-agent contrastive optimization.

Theoretical studies have also examined the foundations of weak-to-strong generalization (Lang et al., 2024; Charikar et al., 2024; Wu and Sahai, 2024), while safety concerns, including risks of deceptive outcomes and backdoor attacks, have been explored (Yang et al., 2024a; Zhao et al., 2024; Ye et al., 2024).

7 Conclusion

In this paper, we introduce a two-stage training framework to enhance weak-to-strong generalization through mitigating overfitting. By focusing on both supervision and question quality, we demonstrate that traditional data filtering methods, while improving supervision, can reduce question difficulty and diversity. Our framework mitigates this by relabeling discarded questions using the fine-tuned strong model, maintaining both supervision accuracy and question quality.

Experiments on the GSM8k and MATH benchmarks demonstrate that our approach significantly outperforms conventional weak-to-strong generalization methods, improving the performance gap recovered (PGR). This validates the effectiveness of our framework in addressing overfitting and enhancing model capabilities on challenging tasks.

Limitations

Our experiments demonstrate strong performance on mathematical reasoning tasks, though the framework's effectiveness remains to be validated across other domains. Through extensive experimentation, we identified optimal confidence thresholds for filtering model predictions. However, these thresholds vary significantly across different tasks and datasets, making automatic threshold selection an important direction for future research. Additionally, the computational overhead of our two-stage finetuning approach, particularly in the second stage, may pose scalability challenges for large-scale applications or real-time scenarios.

References

Aakriti Agrawal, Mucong Ding, Zora Che, Chenghao Deng, Anirudh Satheesh, John Langford, and Furong Huang. 2024. Ensemw2s: Can an ensemble of llms be leveraged to obtain a stronger llm? *Preprint*, arXiv:2410.04571.

Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, Huazuo Gao, Kaige Gao, Wenjun Gao, Ruiqi Ge, Kang Guan, Daya Guo, Jianzhong Guo, Guangbo Hao, Zhewen Hao, Ying He, Wenjie Hu, Panpan Huang, Erhang Li, Guowei Li, Jiashi Li, Yao Li, Y. K. Li, Wenfeng Liang, Fangyun Lin, Alex X. Liu, Bo Liu, Wen Liu, Xiaodong Liu, Xin Liu, Yiyuan Liu, Haoyu Lu, Shanghao Lu, Fuli Luo, Shirong Ma, Xiaotao Nie, Tian Pei, Yishi Piao, Junjie Qiu, Hui Qu, Tongzheng Ren, Zehui Ren, Chong Ruan, Zhangli Sha, Zhihong Shao, Junxiao Song, Xuecheng Su, Jingxiang Sun, Yaofeng Sun, Minghui Tang, Bingxuan Wang, Peiyi Wang, Shiyu Wang, Yaohui Wang, Yongji Wang, Tong Wu, Y. Wu, Xin Xie, Zhenda Xie, Ziwei Xie, Yiliang Xiong, Hanwei Xu, R. X. Xu, Yanhong Xu, Dejian Yang, Yuxiang You, Shuiping Yu, Xingkai Yu, B. Zhang, Haowei Zhang, Lecong Zhang, Liyue Zhang, Mingchuan Zhang, Minghua Zhang, Wentao Zhang, Yichao Zhang, Chenggang Zhao, Yao Zhao, Shangyan Zhou, Shunfeng Zhou, Qihao Zhu, and Yuheng Zou. 2024. Deepseek LLM: scaling open-source language models with longtermism. *CoRR*, abs/2401.02954.

Collin Burns, Pavel Izmailov, Jan Hendrik Kirchner, Bowen Baker, Leo Gao, Leopold Aschenbrenner, Yining Chen, Adrien Ecoffet, Manas Joglekar, Jan Leike, Ilya Sutskever, and Jeff Wu. 2023. Weak-to-strong generalization: Eliciting strong capabilities with weak supervision. *Preprint*, arXiv:2312.09390.

Moses Charikar, Chirag Pabbaraju, and Kirankumar Shiragur. 2024. Quantifying the gain in weak-to-strong generalization. *CoRR*, abs/2405.15116.

Wei Chen, Zhen Huang, Liang Xie, Binbin Lin, Houqiang Li, Le Lu, Xinmei Tian, Deng Cai, Yonggang Zhang, Wenxiao Wan, Xu Shen, and Jieping Ye. 2024. From yes-men to truth-tellers: Addressing sycophancy in large language models with pinpoint tuning. *Preprint*, arXiv:2409.01658.

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

Ajeya Cotra. 2021. Why ai alignment could be hard with modern deep learning. Blog post on Cold Takes. Accessed: 2023-09-28.

Ziyun Cui, Ziyang Zhang, Wen Wu, Guangzhi Sun, and Chao Zhang. 2024. Bayesian weaks-to-strong from text classification to generation. *Preprint*, arXiv:2406.03199.

Weilong Dong, Xinwei Wu, Renren Jin, Shaoyang Xu, and Deyi Xiong. 2024. Contrans: Weak-to-strong alignment engineering via concept transplantation. *Preprint*, arXiv:2405.13578.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurélien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, and et al. 2024. The llama 3 herd of models. *CoRR*, abs/2407.21783.

Yue Guo and Yi Yang. 2024. Improving weak-to-strong generalization with reliability-aware alignment. *CoRR*, abs/2406.19032.

Peter Hase, Mohit Bansal, Peter Clark, and Sarah Wiegreffe. 2024. The unreasonable effectiveness of easy training data for hard tasks. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 7002–7024. Association for Computational Linguistics.

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*.

Hunter Lang, David A. Sontag, and Aravindan Vijayaraghavan. 2024. Theoretical analysis of weak-to-strong generalization. *CoRR*, abs/2405.16043.

Jan Leike. 2022. What is the alignment problem?

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. Let's verify step by step. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Yuejiang Liu and Alexandre Alahi. 2024. Co-supervised learning: Improving weak-to-strong generalization with hierarchical mixture of experts. *CoRR*, abs/2402.15505.

Yougang Lyu, Lingyong Yan, Zihan Wang, Dawei Yin, Pengjie Ren, Maarten de Rijke, and Zhaochun Ren. 2024. Macpo: Weak-to-strong alignment via multi-agent contrastive preference optimization. *Preprint*, arXiv:2410.07672.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Ethan Perez, Sam Ringer, Kamile Lukosiute, Karina Nguyen, Edwin Chen, Scott Heiner, Craig Pettit, Catherine Olsson, Sandipan Kundu, Saurav Kadavath, Andy Jones, Anna Chen, Benjamin Mann, Brian Israel, Bryan Seethor, Cameron McKinnon, Christopher Olah, Da Yan, Daniela Amodei, Dario Amodei, Dawn Drain, Dustin Li, Eli Tran-Johnson, Guro Khundadze, Jackson Kernion, James Landis, Jamie Kerr, Jared Mueller, Jeeyoon Hyun, Joshua Landau, Kamal Ndousse, Landon Goldberg, Liane Lovitt, Martin Lucas, Michael Sellitto, Miranda Zhang, Neerav Kingsland, Nelson Elhage, Nicholas Joseph, Noemí Mercado, Nova DasSarma, Oliver Rausch, Robin Larson, Sam McCandlish, Scott Johnston, Shauna Kravec, Sheer El Showk, Tamera Lanham, Timothy Telleen-Lawton, Tom Brown, Tom Henighan, Tristan Hume, Yuntao Bai, Zac Hatfield-Dodds, Jack Clark, Samuel R. Bowman, Amanda Askell, Roger Grosse, Danny Hernandez, Deep Ganguli, Evan Hubinger, Nicholas Schiefer, and Jared Kaplan. 2023. Discovering language model behaviors with model-written evaluations. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 13387–13434. Association for Computational Linguistics.

Mrinank Sharma, Meg Tong, Tomasz Korbak, David Duvenaud, Amanda Askell, Samuel R. Bowman, Esin Durmus, Zac Hatfield-Dodds, Scott R. Johnston, Shauna Kravec, Timothy Maxwell, Sam McCandlish, Kamal Ndousse, Oliver Rausch, Nicholas Schiefer, Da Yan, Miranda Zhang, and Ethan Perez. 2024. Towards understanding sycophancy in language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Anthony Sicilia, Mert Inan, and Malihe Alikhani. 2024. Accounting for sycophancy in language model uncertainty estimation. *Preprint*, arXiv:2410.14746.

Jerry W. Wei, Da Huang, Yifeng Lu, Denny Zhou, and Quoc V. Le. 2023. Simple synthetic data reduces sycophancy in large language models. *CoRR*, abs/2308.03958.

Jiaxin Wen, Ruiqi Zhong, Akbir Khan, Ethan Perez, Jacob Steinhardt, Minlie Huang, Samuel R. Bowman, He He, and Shi Feng. 2024. Language models learn to mislead humans via RLHF. *CoRR*, abs/2409.12822.

David X. Wu and Anant Sahai. 2024. Provable weak-to-strong generalization via benign overfitting. *Preprint*, arXiv:2410.04638.

Wenkai Yang, Shiqi Shen, Guangyao Shen, Zhi Gong, and Yankai Lin. 2024a. Super(ficial)-alignment: Strong models may deceive weak models in weak-to-strong generalization. *CoRR*, abs/2406.11431.

Yuqing Yang, Yan Ma, and Pengfei Liu. 2024b. Weak-to-strong reasoning. *CoRR*, abs/2407.13647.

Ruimeng Ye, Yang Xiao, and Bo Hui. 2024. Weak-to-strong generalization beyond accuracy: a pilot study in safety, toxicity, and legal reasoning. *Preprint*, arXiv:2410.12621.

Shuai Zhao, Leilei Gan, Zhongliang Guo, Xiaobao Wu, Luwei Xiao, Xiaoyu Xu, Cong-Duy Nguyen, and Luu Anh Tuan. 2024. Weak-to-strong backdoor attack for large language models. *Preprint*, arXiv:2409.17946.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. 2024. Llamafactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, Bangkok, Thailand. Association for Computational Linguistics.

A Dataset details

A.1 Dataset Statistics

For the original question set used in GSM8K and MATH, we followed the methodology of Yang et al. (2024b), adopting the same training set for both datasets. Specifically, we used their dataset D_2, which was employed for training the Llama 2 70B model. For GSM8K, the dataset consists of 7,000 samples, while for MATH, the dataset comprises 6,000 samples.

For evaluation, we utilized the original evaluation set for GSM8K and the test set from Lightman et al. (2024), which contains 500 samples. We compared the model's performance on the 500 samples subset with that on the original test dataset, with details provided in Appendix C.2.

A.2 Implementation Details

For answer generation within the framework, we utilize chain-of-thought (CoT) prompting, as its necessity has been outlined in Section 5.4. In Stage I, answers are generated using zero-shot CoT prompting for the weak models in the Deepseek series. However, for the Llama 3 series, we observed that the Llama 3 9B Instruct model performed below expectations, prompting us to switch from zero-shot to one-shot CoT prompting to enhance its performance.

For sampling parameters, we generate answers with a temperature of 0.6 and top-p of 0.9 for uncertainty-based filtering to ensure diverse and coherent outputs, while using greedy decoding during evaluation to enhance stability.

In both Stage II and the experimental Stage Exp, discussed in Section 5.5, all answers are generated using zero-shot prompting. During the filtering process, after excluding answers based on model confidence, we also discard responses that fail to generate valid answers or do not adhere to the CoT format.

B Training Details

For the supervised finetuning in our framework, we perform full-parameter finetuning on the strong model. The finetuning is carried out with a learning rate of 110^{-5}, a warmup ratio of 0.1, and a cosine learning rate scheduler. We use a batch size of 128 and train for 2 epochs on both the GSM8K and MATH datasets. The implementation is based on the LlamaFactory (Zheng et al., 2024) framework and all experiments are conducted using 64 H100 80GB GPUs to ensure efficient processing and model optimization.

C Additional Analysis

C.1 The Role of Chain-of-Thought in Weak-to-Strong Reasoning

In contrast to the original weak-to-strong generalization framework proposed by (Burns et al., 2023), where all tasks are classification-based, reasoning tasks like GSM8K and MATH consist of open-ended questions that lack definitive answer sets. Previous work has utilized chain-of-thought prompting to enhance performance (Guo and Yang, 2024; Yang et al., 2024b). This raises the question: **Can weak-to-strong generalization remain effective without chain-of-thought prompting?**

To explore this, we replicate the same baseline settings, comparing using chain-of-thought answers to manually constructed direct answers. The results are shown in Table 3.

	Chain-of-Thought	Direct Answer
GSM8K		
Weak Model	74.8	14.6
Strong Ceiling	80.36	30.93
Weak-to-Strong	75.2	13.64
PGR	7.19%	-5.87%(-13.06%)
MATH		
Weak Model	23.8	14.6
Strong Ceiling	33.2	30.93
Weak-to-Strong	27.2	11.4
PGR	36.17%	-31.8%(-76.97%)

Table 3: Performance comparison between chain-of-thought and direct answer approaches in weak-to-strong generalization on GSM8K and MATH datasets with Deepseek series.

When omitting chain-of-thought prompting, we fail to observe generalization in strong models, as finetuned strong models perform worse than their weak teachers. This can be attributed to the fact that chain-of-thought prompting facilitates step-by-step reasoning, which is critical for the strong model to learn from the weak model. It enables the strong model to verify whether each step is correct or incorrect and learn how to break down the whole question into smaller steps. In contrast, the direct answer approach may mislead the model due to the lack of reasoning paths, while incorrect labels may cause more harm than using chain-of-thought, as strong model can learn nothing but false results. We conclude that for reasoning tasks within weak-to-strong generalization, chain-of-thought prompting significantly aids the learning process. Moreover, it may prove beneficial in other tasks and areas under weak-to-strong generalization.

C.2 Is MATH 500 Precise Enough Compared to MATH 5000?

As introduced in Section 2, the Performance Gap Recovered (PGR) metric quantifies the effectiveness of weak-to-strong generalization by comparing the performances of three models: the weak model, strong ceiling model, and finetuned strong model. Our initial evaluations used a subset of 500 test samples (MATH500). Given this relatively small sample size, performance variations of up to 0.2 points per test sample were observed. This variation could be particularly significant when the performance gap between weak and strong ceiling models is small, potentially affecting the reliability of our results.

To validate our findings, we conducted additional evaluations on the complete test set (MATH5000) using models from the DeepSeek series. The results are presented in Table 4.

Model	MATH500	MATH5000
Weak Model	11.4	9.34
Strong Ceiling	19.6	20.12
Stage I Models		
Stage I-Threshold-30%	21.2 (119.51%)	19.96 (98.52%)
Stage I-Threshold-40%	19.6 (100.00%)	17.58 (76.44%)
Stage I-Threshold-50%	17.6 (75.61%)	16.84 (69.57%)
Stage II Models		
Stage I-30% + Stage II-30%	21.4 (121.95%)	21.3 (110.95%)
Stage I-30% + Stage II-40%	21.8 (126.83%)	20.9 (107.24%)
Stage I-30% + Stage II-50%	19.4 (97.56%)	19.48 (94.06%)
Stage I-40% + Stage II-30%	20.4 (109.76%)	19.62 (95.36%)
Stage I-40% + Stage II-40%	19.8 (102.44%)	19.46 (93.88%)
Stage I-40% + Stage II-50%	17.4 (73.17%)	17.62 (76.81%)
Stage I-50% + Stage II-30%	20.6 (112.20%)	19.98 (98.70%)
Stage I-50% + Stage II-40%	20.6 (112.20%)	20.5 (103.53%)
Stage I-50% + Stage II-50%	19.4 (97.56%)	18.8 (87.76%)
Stage I-50% + Stage II-60%	18.6 (87.80%)	18.38 (83.86%)

Table 4: Performance comparison between MATH500 and MATH5000 test sets. Numbers in parentheses represent PGR values.

The results in Table 4 demonstrate that our framework achieves consistent performance across both MATH500 and MATH5000. While the absolute accuracy values remain similar, the slightly lower PGR on MATH5000 can be attributed to the weaker baseline performance of the weak model. However, this difference does not significantly impact our framework's effectiveness. These findings confirm that MATH500 serves as a reliable representative subset for evaluating model performance using PGR, and our framework maintains its effectiveness for weak-to-strong reasoning across different evaluation scales.

D Additional Experimental Results

D.1 Detailed Analysis of Section Diversity Shifts

In this appendix, we provide a comprehensive analysis of the changes in section distribution across filtering thresholds for both Stage I and Stage II of our framework. As shown in Figure 8 for Stage I, increasing the filtering threshold leads to a noticeable reduction in several minor categories, negatively impacting the strong model's ability to generalize effectively across a diverse range of topics. Similarly, in Stage II, as depicted in Figure 9 for Llama 3 MATH (Stage I-Threshold-70%), we observe a recovery in certain minor categories, highlighting the delicate balance between filtering for improved accuracy and maintaining the diversity necessary for robust generalization. Detailed breakdowns of these shifts are provided to offer a clearer understanding of how the filtering process influences the training data distribution across various mathematical categories.

D.2 Numeric Results of All Models and Datasets

We present the numerical results for all models and datasets used in the experiments. It includes performance metrics for different configurations across the GSM8K and MATH benchmarks, showcasing the impact of various stages and filtering thresholds on model performance.

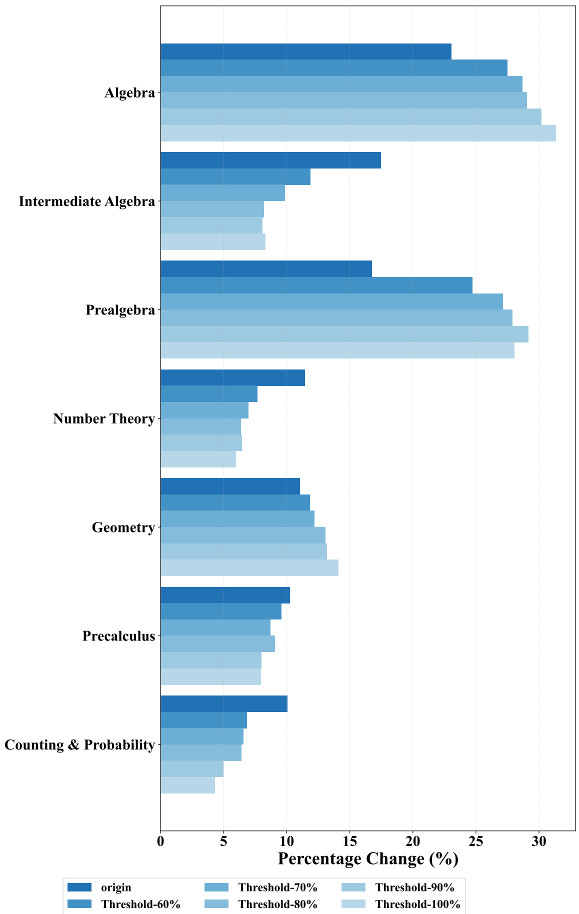


Figure 8: Changes in topic distribution across filtering thresholds for all mathematical categories in Stage I.(Llama3 MATH) Filtering causes shifts in topic distribution, with minor categories seeing more reductions.

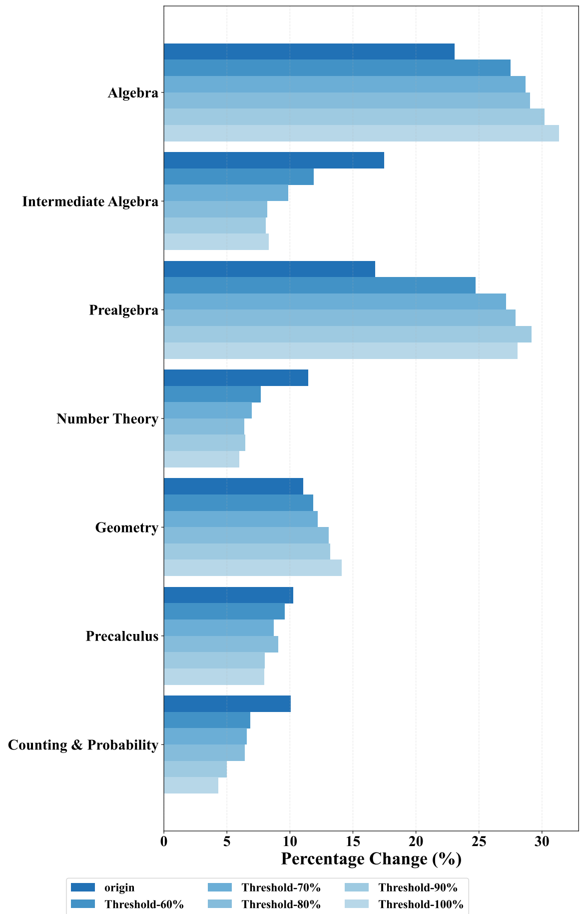


Figure 9: Changes in topic distribution across filtering thresholds for all mathematical categories in Stage II.(LLama3 MATH Stage I-Threshold-70%) We observe recovery in several minor categories, while sections including algebra, intermediate algebra, prealgebra are also effected by difficulty.

	Accuracy	Performance gap recovered(PGR)
Basic Settings		
Weak Model	74.8%	0%
Strong Ceiling	80.36%	100%
Conventional Weak-to-Strong	75.2%	7.19%
Stage I		
Stage I-Threshold-30%	79.37%	82.19%
Stage I-Threshold-40%	79.51%	84.71%
Stage I-Threshold-50%	78.99%	75.36%
Stage I-Threshold-60%	80.07%	94.78%
Stage I-Threshold-70%	80.28%	98.56%
Stage I-Threshold-80%	80.06%	94.60%
Stage I-Threshold-90%	80.13%	95.86%
Stage I-Threshold-100%	78.16%	60.43%
Stage II based on Stage I Threshold-50%		
Stage I-50% + Stage II-50%	80.28%	98.56%
Stage I-50% + Stage II-60%	80.89%	109.53%
Stage I-50% + Stage II-70%	79.62%	86.69%
Stage I-50% + Stage II-80%	79.37%	82.19%
Stage II based on Stage I Threshold-60%		
Stage I-60% + Stage II-50%	80.28%	98.56%
Stage I-60% + Stage II-60%	81.50%	120.50%
Stage I-60% + Stage II-70%	81.04%	112.23%
Stage I-60% + Stage II-80%	81.34%	117.63%
Stage II based on Stage I Threshold-70%		
Stage I-70% + Stage II-60%	80.89%	109.53%
Stage I-70% + Stage II-70%	80.36%	100.00%
Stage I-70% + Stage II-80%	81.19%	114.93%
Stage I-70% + Stage II-90%	80.89%	109.53%
Stage II based on Stage I Threshold-80%		
Stage I-80% + Stage II-70%	80.43%	101.26%
Stage I-80% + Stage II-80%	80.33%	99.46%
Stage I-80% + Stage II-90%	80.45%	101.62%
Stage I-80% + Stage II-100%	80.74%	106.83%

Table 5: Llama3 GSM8k

	Accuracy	Performance gap recovered(PGR)
Basic Settings		
Weak Model	23.8%	0%
Strong Ceiling	33.2%	100%
Conventional Weak-to-Strong	27.2%	36.17%
Stage I		
Stage I-Threshold-30%	27.2%	36.17%
Stage I-Threshold-40%	29.8%	63.83%
Stage I-Threshold-50%	30.0%	65.96%
Stage I-Threshold-60%	31.4%	80.85%
Stage I-Threshold-70%	34.4%	112.77%
Stage I-Threshold-80%	33.2%	100.00%
Stage I-Threshold-90%	32.6%	93.62%
Stage I-Threshold-100%	22.6%	-12.77%
Stage II based on Stage I Threshold-60%		
Stage I-60% + Stage II-50%	27.0%	34.04%
Stage I-60% + Stage II-60%	30.6%	72.34%
Stage I-60% + Stage II-70%	32.4%	91.49%
Stage I-60% + Stage II-80%	32.4%	91.49%
Stage I-60% + Stage II-90%	29.0%	55.32%
Stage I-60% + Stage II-100%	30.7%	73.40%
Stage II based on Stage I Threshold-70%		
Stage I-70% + Stage II-60%	32.2%	89.36%
Stage I-70% + Stage II-70%	32.4%	91.49%
Stage I-70% + Stage II-80%	35.2%	121.28%
Stage I-70% + Stage II-90%	34.2%	110.64%
Stage I-70% + Stage II-100%	33.2%	100.00%
Stage II based on Stage I Threshold-80%		
Stage I-80% + Stage II-70%	30.0%	65.96%
Stage I-80% + Stage II-80%	32.2%	89.36%
Stage I-80% + Stage II-90%	33.8%	106.38%
Stage I-80% + Stage II-100%	32.8%	95.74%

Table 6: Llama 3 MATH

Model	Accuracy	Performance gap recovered(PGR)
Basic Settings		
Weak Model	48.36%	0%
Strong Ceiling	75.66%	100%
conventional Weak-to-Strong	62.39%	51.39%
Stage I		
Stage I-Threshold-30%	68.68%	74.43%
Stage I-Threshold-40%	70.96%	82.78%
Stage I-Threshold-50%	69.74%	78.32%
Stage I-Threshold-60%	70.35%	80.55%
Stage I-Threshold-70%	71.11%	83.33%
Stage I-Threshold-80%	69.14%	76.12%
Stage I-Threshold-90%	68.38%	73.33%
Stage I-Threshold-100%	67.55%	70.29%
Stage II based on Stage I Threshold-40%		
Stage I-40% + Stage II-30%	72.63%	88.90%
Stage I-40% + Stage II-40%	72.32%	87.77%
Stage I-40% + Stage II-50%	70.58%	81.39%
Stage I-40% + Stage II-60%	72.17%	87.22%
Stage II based on Stage I Threshold-60%		
Stage I-60% + Stage II-60%	70.28%	80.29%
Stage I-60% + Stage II-70%	71.49%	84.73%
Stage I-60% + Stage II-80%	70.28%	80.29%
Stage I-60% + Stage II-90%	70.28%	80.29%
Stage II based on Stage I Threshold-70%		
Stage I-70% + Stage II-60%	72.40%	88.06%
Stage I-70% + Stage II-70%	72.94%	90.04%
Stage I-70% + Stage II-80%	71.64%	85.27%
Stage I-70% + Stage II-90%	72.55%	88.61%
Stage II based on Stage I Threshold-80%		
Stage I-80% + Stage II-70%	70.20%	80.00%
Stage I-80% + Stage II-80%	70.50%	81.10%
Stage I-80% + Stage II-90%	71.47%	84.65%
Stage I-80% + Stage II-100%	70.35%	80.55%

Table 7: Deepseek-GSM8K

Model	Accuracy	Performance gap recovered(PGR)
Basic Settings		
Weak Model	11.4%	0%
Strong Ceiling	19.6%	100%
conventional Weak-to-Strong	16.8%	65.85%
Stage I		
Stage I-Threshold-30%	21.2%	119.51%
Stage I-Threshold-40%	19.6%	100.00%
Stage I-Threshold-50%	17.6%	75.61%
Stage I-Threshold-60%	15.8%	53.66%
Stage I-Threshold-70%	16.4%	60.98%
Stage I-Threshold-80%	15.0%	43.90%
Stage I-Threshold-90%	12.0%	7.32%
Stage II based on Threshold-30%		
Stage I-30% + Stage II-30%	21.4%	121.95%
Stage I-30% + Stage II-40%	21.8%	126.83%
Stage I-30% + Stage II-50%	19.4%	97.56%
Stage I-30% + Stage II-60%	19.2%	95.12%
Stage I-30% + Stage II-70%	19.0%	92.68%
Stage II based on Threshold-40%		
Stage I-40% + Stage II-30%	20.4%	109.76%
Stage I-40% + Stage II-40%	19.8%	102.44%
Stage I-40% + Stage II-50%	17.4%	73.17%
Stage I-40% + Stage II-60%	18.0%	80.49%
Stage II based on Threshold-50%		
Stage I-50% + Stage II-30%	20.6%	112.20%
Stage I-50% + Stage II-40%	20.6%	112.20%
Stage I-50% + Stage II-50%	19.4%	97.56%
Stage I-50% + Stage II-60%	18.6%	87.80%

Table 8: Deepseek-MATH