Extended Abstract

Motivation On-policy distillation (OPD) has emerged as an effective approach for transferring capabilities from specialized teacher language models to smaller student models. Recent advances such as G-OPD further improve distillation by extrapolating beyond the teacher rather than merely imitating it. However, existing OPD methods rely exclusively on feedback from teachers that are aligned with the target task and largely ignore signals from other available teachers. In practice, specialized teachers often exhibit strong domain-specific biases: a teacher that performs well on one task may perform poorly on unrelated tasks. We hypothesize that such misaligned teachers can provide useful negative feedback by revealing behaviors that should be discouraged for the target task.

Method We propose Teacher-Contrastive On-Policy Distillation (TC-OPD), a multi-teacher extension of OPD that incorporates both expert and anti-expert feedback. Building on the teacher extrapolation framework of G-OPD, our method simultaneously extrapolates toward an aligned expert teacher and away from a misaligned anti-expert teacher. Intuitively, the expert teacher provides positive learning signals describing what the student should do, while the anti-expert provides contrastive signals describing what the student should avoid. This enables the student to exploit teacher information that is discarded by existing expert-only approaches.

Implementation We implement TC-OPD by extending the G-OPD framework with an additional teacher model. During training, we collect student rollouts and evaluate them under both the expert and anti-expert teachers. The resulting teacher scores are combined through a teacher-contrastive extrapolation objective. To isolate the effect of contrastive feedback, we consider a controlled single-task setting where mathematical reasoning is the target task, a math-specialized model serves as the expert teacher, and a coding-specialized model serves as the anti-expert. All experiments are implemented using the `verl` framework on 8 NVIDIA A100 GPUs.

Results We evaluate TC-OPD on the AIME24 and AIME25 mathematical reasoning benchmarks. Compared with standard G-OPD, moderate anti-expert extrapolation improves final performance and achieves the best result on AIME24. We additionally observe faster learning during training, indicating that teacher-contrastive feedback can improve optimization efficiency. Interestingly, overly aggressive anti-expert extrapolation degrades performance, suggesting that the signals provided by the expert and anti-expert must be balanced carefully.

Our qualitative analysis shows that the coding anti-expert exhibits reasoning patterns that differ substantially from those of the mathematical expert, supporting the hypothesis that specialization induces domain-specific preferences. Consistent with this observation, TC-OPD encourages the student to move away from the verbose reasoning style exhibited by the coding anti-expert. This provides evidence that anti-expert feedback can shape not only task performance but also the reasoning behavior learned by the student.

Discussion Our experiments focus on a controlled single-task distillation setting. An important direction for future work is to evaluate TC-OPD in multi-task settings, where the student must simultaneously acquire capabilities from multiple specialized teachers. In such settings, knowledge interference and catastrophic forgetting become important concerns, and it remains unclear whether teacher-contrastive feedback alleviates or exacerbates these effects. Understanding this trade-off is an interesting direction for future research.

In addition, while TC-OPD incurs only a modest runtime overhead, its memory consumption grows with the number of teachers. Future work could explore distributed implementations that shard teachers across GPUs, reducing the memory overhead of multi-teacher distillation.

Conclusion We demonstrated that such feedback can be transformed into useful learning signals through teacher-contrastive extrapolation. By simultaneously extrapolating toward expert teachers and away from anti-experts, TC-OPD extends the teacher extrapolation framework beyond expert-only supervision. Our results suggest that teacher-contrastive distillation is a promising direction for multi-teacher post-training and motivate future work on multi-task distillation and scalable multi-teacher training systems.

Teacher-Contrastive On-Policy Distillation

Juntong Shi
Department of Computer Science
Stanford University
juntong@stanford.edu

Abstract

On-policy distillation (OPD) transfers capabilities from specialized teacher language models to smaller student models by training on student-generated trajectories with teacher feedback. Recent work has shown that teacher extrapolation can further improve distillation by amplifying teacher-preferred behaviors. However, existing methods rely exclusively on expert-aligned feedback and ignore potentially useful information from other available teachers. In this work, we propose Teacher-Contrastive On-Policy Distillation (TC-OPD), a multi-teacher extension of OPD that simultaneously extrapolates toward an aligned expert teacher and away from a misaligned anti-expert teacher. We evaluate TC-OPD on mathematical reasoning tasks using a math teacher as the expert and a coding teacher as the anti-expert. Experimental results show that moderate anti-expert extrapolation improves performance and learning efficiency over standard G-OPD while introducing only modest computational overhead. These findings suggest that feedback from misaligned teachers can provide useful learning signals and highlight teacher-contrastive distillation as a promising direction for multi-teacher post-training.

1 Introduction

Post-training has become a critical component of modern large language models (LLMs), enabling strong capabilities in reasoning, instruction following, and other specialized tasks. Among recent post-training techniques, On-Policy Distillation (OPD) (Lu and Lab, 2025) has emerged as an effective approach for transferring capabilities from a large specialized teacher to a smaller student model. Unlike supervised fine-tuning (SFT), which trains the student on teacher-generated trajectories, OPD trains on the student's own trajectories and uses the teacher to provide corrective feedback. This approach improves distillation efficiency and stability and has become an increasingly important component of modern post-training pipelines (Yang et al., 2025b; DeepSeek-AI, 2026).

Existing OPD methods primarily focus on distillation from a single teacher. More recently, multi-teacher distillation has been explored, where multiple specialized teachers provide feedback on their corresponding domains. In practice, these approaches (Yang et al., 2025b; DeepSeek-AI, 2026; Yang et al., 2026) use only expert-aligned feedback and largely ignore feedback from teachers that is misaligned with the current task (Figure 1). However, such feedback is both inexpensive to obtain and potentially informative. Due to specialization, a teacher that performs well in one domain often performs worse in unrelated domains. Consequently, its preferences may reveal behaviors that should be discouraged for the target task, allowing it to serve as an informative anti-expert that provides negative learning signals. Furthermore, these signals are inexpensive to acquire in practice, since scoring responses can be performed in parallel, whereas rollout generation is sequential and typically dominates the cost of OPD.

Motivated by this observation, we propose TC-OPD (**T**eacher-**C**ontrastive **O**n-**P**olicy **D**istillation), a multi-teacher extension of OPD that incorporates both expert and anti-expert feedback. Building on the teacher extrapolation framework of G-OPD (Yang et al., 2026), our method simultaneously

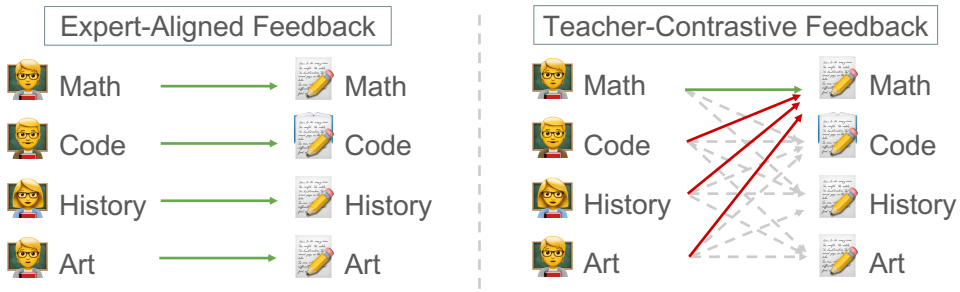

Figure 1: Compare Expert-Aligned Feedbak in the single tearcher distillation settig and Teacher-Contrastive Feedback in the multi-teacher distillation setting.

extrapolates toward an aligned expert teacher and away from a misaligned anti-expert teacher. This allows the student to leverage teacher-contrastive signals that are discarded by existing expert-only approaches.

We evaluate TC-OPD in a controlled mathematical reasoning setting using a math teacher as the expert and a coding teacher as the anti-expert. Experimental results show that moderate anti-expert extrapolation improves both final performance and learning efficiency over standard G-OPD, while introducing only a modest computational overhead. These findings suggest that feedback from misaligned teachers can provide useful learning signals and that teacher-contrastive distillation is a promising direction for multi-teacher post-training.

2 Related Work

On-policy distillation. On-policy distillation (OPD) (Lu and Lab, 2025) has emerged as an effective approach for transferring knowledge from a specialized teacher language model to a small, non-specialized student model. It is increasingly used in post-training pipelines for modern language models (Yang et al., 2025b; DeepSeek-AI, 2026). Unlike traditional supervised fine-tuning (SFT), which trains the student on full teacher-generated trajectories, OPD conditions training on the student's self-generated trajectories and uses the teacher to provide correction signals. This shifts the training distribution from the teacher's behavior to the student's own behavior, which makes the student more robust to compounding error. Conceptually, this distinction resembles the difference between behavior cloning and DAgger (Ross et al., 2011) in imitation learning.

Teacher-extrapolated on-policy distillation. Generalized On-Policy Distillation (G-OPD) (Yang et al., 2026) extends vanilla OPD by allowing the student to extrapolate beyond the teacher. Specifically, G-OPD replaces the original teacher distillation target with a teacher-extrapolated Product-of-Experts target constructed from the teacher and a reference model. This target amplifies teacher-preferred behaviors, so the student is encouraged not only to match the teacher but also to move further in the direction favored by the teacher. Our method builds directly on this extrapolation framework and extends it to incorporate contrastive feedback from multiple teachers.

Multi-teacher Distillation. Recently, OPD has been extended to the multi-teacher setting, where each teacher specializes in a different domain and the student is trained to acquire all of their capabilities simultaneously. Prior approaches (Team et al., 2026; DeepSeek-AI, 2026; Yang et al., 2026) typically use each teacher only on its corresponding domain, which can be viewed as performing multiple single-task OPD at the same time. Compared to our method, it failed to explore potentially useful learning signals from feedback of misaligned teachers.

More complex approaches (Yuan et al., 2020; Yang et al., 2025c) use routing or learned weighting to route each input to the most relevant set of effective teachers. While these methods can combine signals from different teachers, they rely on indirect signals derived from the input rather than directly leveraging the information in teacher predictions. Furthermore, these approaches are often evaluated on simple classification or prediction tasks, and it remains unclear how well they extend to open-ended generation tasks.

3 Preliminaries

3.1 Problem Setup

Let π_b denote a base student language model and $\{\pi_k\}_{k=1}^{K}$ denote a collection of specialized teacher models. Each teacher is associated with a task domain $\mathcal{D}_k$, such as mathematical reasoning or code generation. Our goal is to transfer the specialized capabilities of the teacher models into the smaller and more efficient student model.

3.2 On-Policy Distillation (OPD)

We briefly review On-Policy Distillation (OPD) (Lu and Lab, 2025), which serves as the foundation of our method. Consider a prompt $x \sim \mathcal{D}$, a response y, a reference policy π_{ref}, and a trainable policy π_θ. In KL-regularized, on-policy reinforcement learning, we want to maximize the objective:

$$J_{\text{RL}}(\theta) = \mathbb{E}_{x\sim\mathcal{D},\, y\sim\pi_\theta(\cdot|x)} \left[r(x,y) - D_{\text{KL}}(\pi_\theta(\cdot|x) \parallel \pi_{\text{ref}}(\cdot|x)) \right]. \tag{1}$$

A well-known result shows that the optimal policy induced by reward r takes the form

$$\pi_r^\star(y|x) = \frac{1}{Z(x)} \pi_{\text{ref}}(y|x) \exp(r(x,y)), \tag{2}$$

where $Z(x)$ is a normalization constant.

OPD follows the same idea. Suppose instead that we are given a target teacher policy π^t and wish to train the student to mimic it. By choosing the reward

$$r(x,y) = \log \frac{\pi^t(y|x)}{\pi_{\text{ref}}(y|x)}, \tag{3}$$

the induced optimal policy becomes exactly the teacher policy:

$$\pi_r^\star(y|x) = \pi^t(y|x). \tag{4}$$

Substituting this reward into the KL-regularized RL objective yields

$$J_{\text{OPD}}(\theta) = \mathbb{E}_{x\sim\mathcal{D},\, y\sim\pi_\theta(\cdot|x)} \left[\log \frac{\pi^t(y|x)}{\pi_{\text{ref}}(y|x)} - \log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)} \right] \tag{5}$$

$$= -\mathbb{E}_{x\sim\mathcal{D}} \left[D_{\text{KL}}\big(\pi_\theta(\cdot|x) \parallel \pi^t(\cdot|x)\big) \right]. \tag{6}$$

Therefore, OPD can be viewed as a form of on-policy reinforcement learning that implicitly minimizes the reverse KL divergence between the student and teacher distributions.

A key feature of OPD is that the expectation is taken over responses sampled from the current student policy, i.e., $y \sim \pi_\theta(\cdot|x)$. Consequently, the teacher is queried on trajectories that are actually visited by the student during training. This contrasts with conventional supervised distillation, which only matches the teacher distribution on trajectories sampled from the teacher itself.

OPD can also be viewed as a language-model instantiation of DAgger (Ross et al., 2011), where teacher feedback on student-generated trajectories helps reduce the compounding errors caused by distribution shift.

3.3 Teacher-Extrapolated OPD

We next review Teacher-Extrapolated OPD, also known as Generalized OPD (G-OPD) (Yang et al., 2026), a single-teacher extension of OPD. While OPD trains the student to match the teacher distribution, it is natural to ask whether the student can surpass the teacher by extrapolating its behavior. To this end, G-OPD replaces the teacher policy π^t with an extrapolated target distribution

$$\pi^\star(y|x) \propto \pi_{\text{ref}}(y|x)^{1-\lambda} \pi^t(y|x)^\lambda, \tag{7}$$

where λ controls the degree of extrapolation. When $\lambda = 1$, the target reduces to the original teacher policy and recovers OPD. When $0 < \lambda < 1$, the target interpolates between the reference and teacher policies. When $\lambda > 1$, the target extrapolates beyond the teacher by further amplifying behaviors

preferred by the teacher relative to the reference model, while actively suppressing behaviors preferred by the reference model relative to the teacher.

Using the reward-tilted policy formulation in Equation (2), the reward corresponding to this target distribution is

$$r(x, y) = \lambda \log \frac{\pi^t(y|x)}{\pi_{\text{ref}}(y|x)}. \tag{8}$$

Substituting this reward into the KL-regularized RL objective yields

$$\begin{aligned} J_{\text{G-OPD}}(\theta) &= \mathbb{E}_{x\sim\mathcal{D},\, y\sim\pi_\theta(\cdot|x)} \left[\lambda \log \frac{\pi^t(y|x)}{\pi_{\text{ref}}(y|x)} - \log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)} \right] \\ &= \mathbb{E}_{x\sim\mathcal{D}} \left[\mathbb{E}_{y\sim\pi_\theta(\cdot|x)} \left[\lambda \log \frac{\pi^t(y|x)}{\pi_{\text{ref}}(y|x)} \right] - D_{\text{KL}}(\pi_\theta(\cdot|x) \parallel \pi_{\text{ref}}(\cdot|x)) \right]. \end{aligned} \tag{9}$$

4 Method

Teacher-Extrapolated OPD considers a single teacher and extrapolates its behavior relative to the reference model. As a result, it relies exclusively on *expert-aligned feedback* and discards information from other available teachers. In practice, however, we often have access to multiple specialized teachers. For example, when distilling mathematical reasoning, we may have access not only to a mathematics specialist but also to a coding specialist. While only the mathematics teacher is directly aligned with the target task, the coding teacher may still provide useful information about behaviors that should be discouraged.

This observation is motivated by the fact that specialization often comes with trade-offs. A teacher that is highly optimized for one domain typically improves in that domain at the expense of performance on unrelated domains. Consequently, a mismatched teacher can reveal behaviors that are less desirable for the target task and therefore serve as an informative anti-expert. Rather than treating such teachers as irrelevant, we propose to incorporate them as a source of *teacher-contrastive feedback*.

Motivated by the observation that teacher feedback can be either supportive or contradictory, we assign each teacher an individual extrapolation coefficient that controls both the direction and strength of extrapolation. We then define the target distribution as a product of teachers extrapolated to different degrees:

$$\pi^\star(y|x) \propto \left[\prod_{k=1}^{K} \pi_{\text{ref}}(y|x)^{1-\lambda_k} \pi_k(y|x)^{\lambda_k} \right]^{1/K}. \tag{10}$$

The sign of λ_k controls the extrapolation direction of teacher k: positive coefficients amplify behaviors preferred by teacher k, whereas negative coefficients suppress them. The magnitude of λ_k determines the strength of extrapolation. The factor $1/K$ normalizes the contribution of the teacher and ensures that when $K = 1$, the formulation reduces to Teacher-Extrapolated OPD.

This formulation naturally allows different teachers to play different roles in the distillation process. Teachers whose expertise aligns with the target task are treated as experts and assigned positive extrapolation coefficients, encouraging the student to move toward their preferred behaviors. Conversely, teachers specialized in other domains are treated as anti-experts and assigned negative extrapolation coefficients, discouraging behaviors they favor. For example, when distilling mathematical reasoning, the mathematics teacher acts as an expert while the coding teacher acts as an anti-expert.

Using the reward-tilted policy formulation in Equation (2) again, the reward corresponding to this target distribution is

$$r(x, y) = \frac{1}{K} \sum_{k=1}^{K} \lambda_k \log \frac{\pi_k(y|x)}{\pi_{\text{ref}}(y|x)}. \tag{11}$$

Finally, substituting this reward function into the KL-regularized RL objective Equation (1) yields the Teacher-Contrastive OPD (TC-OPD) objective

$$J_{\text{TC-OPD}}(\theta) = \mathbb{E}_{x\sim\mathcal{D}} \left[\mathbb{E}_{y\sim\pi_\theta(\cdot|x)} \frac{1}{K} \sum_{k=1}^{K} \lambda_k \left[\log \frac{\pi_k(y|x)}{\pi_{\text{ref}}(y|x)} \right] - D_{\text{KL}}(\pi_\theta(\cdot|x) \,||\, \pi_{\text{ref}}(\cdot|x)) \right]. \tag{12}$$

TC-OPD can be viewed as a strict generalization of Teacher-Extrapolated OPD. When $K = 1$, the objective reduces to the single-teacher setting, whereas for $K > 1$ it allows feedback from multiple specialized teachers to be integrated within a unified extrapolation framework.

5 Experimental Setup

As discussed in the previous sections, our experiments focus on a controlled single-task distillation setting, where the teachers are specialized in different domains but distillation is performed on a single target task. We chose this setting to isolate and better understand the effect of teacher-contrastive feedback. The models we used are relatively small with limited capacity, so multi-task distillation may introduce additional confounding factors such as knowledge interference and forgetting. Evaluating on a single task, therefore, provides a cleaner and more interpretable testbed for analyzing the benefits of TC-OPD.

The experiment setting we used is adapted from Yang et al. (2026), and the single task we focused on is *math reasoning*. Specifically:

Training Datasets. Training Datasets. We use the filtered training datasets released by Yang et al. (2026). For the math domain, they filter the DeepMath (He et al., 2025) dataset and retain 57K samples with difficulty greater than or equal to 6. DeepMath contains challenging competition-level mathematical reasoning problems spanning topics such as algebra, geometry, number theory, and combinatorics. The problems are collected from mathematically verified human-created sources, such as math forums and competition repositories, and are curated through a rigorous filtering pipeline. The corresponding solutions and reasoning traces are generated by LLMs.

Student Model. We use the base Qwen3-4B model (Yang et al., 2025a) as the student. Unlike instruction-tuned or reasoning-enhanced variants, this model has not undergone additional supervised fine-tuning and does not possess an explicit thinking mode. As a result, it provides a clean initialization for subsequent RL training and distillation.

Teacher Models. We reuse the teacher models released by Yang et al. (2026). Both teachers are initialized from the same Qwen3-4B base model as our student and are subsequently trained with reinforcement learning on domain-specific datasets with verifiable rewards. The math teacher is trained with GRPO on DeepMath, which is also the dataset used for our distillation experiments. We therefore designate it as the expert teacher, since its training domain aligns with our target task.

In addition, Yang et al. (2026) trains a coding teacher on the Eurus-RL-Code dataset (Cui et al., 2025), which contains 25K coding problems with verifiable solutions. We use this model as the anti-expert teacher. Intuitively, coding and mathematical reasoning require substantially different capabilities, making the coding teacher a natural source of contrastive feedback for math distillation.

Training Settings. we use a binary reward function: a response receives a reward of 1 if its final answer is correct and 0 otherwise. Building on this setup, we evaluate G-OPD under several expert/anti-expert extrapolation configurations, $(\lambda_e, \lambda_a) \in \{(1.5, 0.0), (1.5, -0.5), (1.5, -1.5)\}$. λ_e denotes the extrapolation factor applied to the expert, and λ_a denotes that applied to the anti-expert. The configuration $(1.5, 0.0)$ corresponds to the original G-OPD, which extrapolates only the expert teacher. In all experiments, the reference model is fixed to the student's initialization, namely the base Qwen3-4B model. We run TC-OPD off-policy, with multiple gradient steps on a single sampple. The imporatance weight is implemented as the simple token-level version.

Our implementation is built on the `verl` framework (Sheng et al., 2024), which provides efficient multi-GPU training for large-scale RL. All experiments are conducted with BF16 precision on 8 NVIDIA A100 GPUs with a training batch size of 128 and a micro-batch size of 1 per GPU. During rollout generation, we dynamically allocate samples until approximately 60% of GPU memory is utilized. Each model is trained for 50 batches. Additional implementation details and hyperparameters can be found in the training scripts provided in our codebase.

Evaluation. For mathematical reasoning evaluation, we consider two challenging competition-level benchmarks: AIME24 (AI-MO, 2024) and AIME25 (OpenCompass, 2025), each containing 30 questions with human-verified solutions. Across all evaluations, we use a sampling temperature of 1.0, top-p of 1.0, and a maximum generation length of 16,384 tokens. For each benchmark problem, we generate 32 independent solutions and compute the average solution accuracy across samples

Table 1: TC-OPD's and baselines' performance at the end of training

Method	AIME24	AIME25
Student	14.17	11.46
Expert	58.00	54.60
Anti-Expert	8.44	6.75
OPD ($\Leftrightarrow$ TC-OPD with $\lambda_e = 1.0, \lambda_a = 0.0$)	60.70	55.60
G-OPD ($\Leftrightarrow$ TC-OPD with $\lambda_e = 1.5, \lambda_a = 0.0$)	65.58	**57.13**
TC-OPD ($\lambda_e = 1.5, \lambda_a = -0.5$)	**67.92**	56.67
TC-OPD ($\lambda_e = 1.5, \lambda_a = -1.5$)	63.94	55.83

(`mean@32`). To determine answer correctness, we employ `Math-Verify`[1], a rule-based mathematical answer verifier.

Baselinse. We compare TC-OPD against the untrained student base, the expert and anti-expert, models trained with OPD and G-OPD.

6 Results

6.1 Quantitative Evaluation

Performance. Table 1 presents the main results on AIME24 and AIME25. As expected, the expert teacher substantially outperforms the student, while the anti-expert performs even worse than the student because it is specialized for a different domain. This confirms that the anti-expert provides a meaningful source of negative feedback.

Comparing the distillation methods, vanilla OPD already recovers and slightly exceeds the expert teacher's performance in our setting. G-OPD further improves upon OPD by extrapolating beyond the expert teacher, demonstrating that teacher extrapolation can transfer useful knowledge that is not directly captured by faithful distillation alone.

Most importantly, TC-OPD matches or exceeds the performance of G-OPD. In particular, the configuration with moderate anti-expert reverse-extrapolation, $(\lambda_e, \lambda_a) = (1.5, -0.5)$, achieves the best performance on AIME24 and remains competitive on AIME25. In contrast, a more aggressive anti-expert penalty, $(1.5, -1.5)$, degrades performance. These results suggest that negative feedback from an anti-expert can provide useful learning signals when applied with an appropriate strength, but excessive suppression may suppress learning useful for the target task.

Learning Efficiency. Beyond final performance, TC-OPD also improves learning efficiency. As shown in Figure 2, the variant with moderate anti-expert extrapolation reaches strong performance in fewer RL steps than G-OPD. This indicates that contrastive feedback not only improves the final policy but can also accelerate the optimization process by providing additional guidance during training.

Computational Cost. Incorporating additional teachers introduces only a modest computational overhead. As shown in Table 2, TC-OPD increases total GPU hours by only 3.5% compared to the single-teacher G-OPD baseline. This overhead remains small because teacher evaluation is inexpensive relative to rollout generation, which dominates the overall training cost.

The primary cost of TC-OPD is increased memory consumption. As shown in Table 2, adding a single anti-expert increases peak GPU memory usage by 8.2 GB. This overhead arises from maintaining an additional 4B-parameter anti-expert model in memory and storing the corresponding teacher logits required by the TC-OPD objective. While runtime scales safely with the number of teachers, additional memory requirements grow approximately linearly, which may become a bottleneck if we want to incorporate many specialized teachers.

[1] `https://github.com/huggingface/Math-Verify`

Table 2: Training cost comparison between the single-teacher G-OPD and multi-teacher TC-OPD.

Method	# Teachers	GPU Hours	Peak GPU Memory (GB)
G-OPD	1	11.7	60.2
TC-OPD	2	12.1	68.4

Table 3: Average response length (in tokens) on the training data.

Coding Anti-Expert	G-OPD	TC-OPD
13789.41	10407.49	9662.02

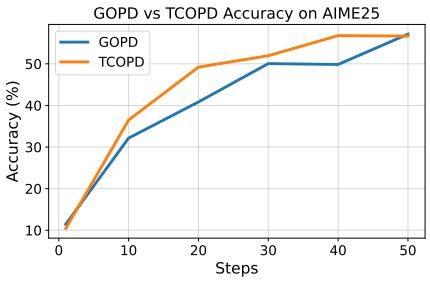

Figure 2: Training reward progression on AIME25 for G-OPD and TC-OPD.

6.2 Qualitative Analysis

We further investigate how TC-OPD affects the learned reasoning behavior beyond the aggregate benchmark scores. Specifically, we analyze the average response length and inspect representative generations from the anti-expert, G-OPD, and TC-OPD.

Response Length. Table 3 compares the average response length of the final models. The coding anti-expert generates the longest responses by a large margin, averaging 13789 tokens. Interestingly, TC-OPD learns an even more concise reasoning style than G-OPD, reducing the average response length by 7.2% while achieving comparable or better benchmark performance. This observation suggests that reverse extrapolation from the anti-expert does not merely affect answer correctness, but also guide the response style learned by the student.

Qualitative Examples. Section A presents representative generations from the anti-expert, G-OPD, and TC-OPD. The coding anti-expert exhibits a noticeably different reasoning style from the math-distilled expert. In particular, it tends to begin by explicitly defining variables, restating all given quantities, and constructing a detailed solution plan before attempting the problem. This behavior likely reflects the instructional style encouraged by training on coding tasks.

In contrast, both G-OPD and TC-OPD learn a reasoning style that is better aligned with mathematical problem solving. They quickly identify the underlying combinatorial structure and proceed directly toward the solution. This observation further supports our choice of using the coding teacher as an anti-expert, since its behavior demonstrates specialization in a domain that differs substantially from the target math task.

The behavioral differences between G-OPD and TC-OPD are more subtle. In the examples we examined, both models produce similar math reasoning patterns, such as the frequent use of words phrases “let’s” and “we should”. We speculate that this is because the strong supervision from the expert teacher dominates the learning signal in our single-task setting. Nevertheless, as shown in Table 3, TC-OPD consistently produces shorter responses than G-OPD, suggesting that anti-expert feedback does influences the learned reasoning style even when the overall ideal teacher solution strategy remains unchanged.

7 Discussion

Multi-Task Distillation. Our experiments focus on a controlled single-task distillation setting, where the target task is mathematical reasoning and the additional teacher serves as an anti-expert from a different domain. While this setting allows us to isolate the effect of teacher-contrastive feedback, it does not fully capture the multi-task scenarios in which multiple specialized teachers are often available. An important direction for future work is to evaluate TC-OPD in a multi-task distillation setting, where the student must simultaneously acquire knowledge from several teachers across different domains.

The multi-task setting introduces additional challenges that’s not considered in this work. In particular, the student must maintain multiple capabilities at the same time. Knowledge interference and forgetting then become important concerns. It remains unclear whether teacher-contrastive feedback alleviates or exacerbates these issues. On one hand, anti-expert feedback may help the student better

isolate domain-specific behaviors and reduce interference between tasks. On the other hand, excessive extrapolation away from one teacher could potentially suppress useful knowledge that overlaps with other domains. Understanding these trade-offs is an interesting direction for future research.

Scalability. Another limitation of the current implementation is its memory efficiency. In our implementation, each training worker hosts all teacher models, causing memory consumption to scale approximately linearly with the number of teachers. While the runtime overhead remains modest because teacher evaluation is inexpensive relative to rollout generation, memory requirements may become a bottleneck when incorporating a large number of specialized teachers. Future work could explore more efficient distributed implementations that shard teacher models across GPUs rather than replicating them on every worker. Such approach could reduce the additional memory overhead from $O(\#\text{teachers})$ to approximately $O(\#\text{teachers}/\#\text{GPUs})$, making TC-OPD more scalable.

8 Conclusion

In this project, we studied whether feedback from misaligned teachers can be leveraged to improve on-policy distillation. We demonstrated that such feedback can be transformed into useful learning signals through teacher-contrastive extrapolation. By simultaneously extrapolating toward expert teachers and away from anti-experts, TC-OPD extends the teacher extrapolation framework beyond expert-only supervision. Our results suggest that teacher-contrastive distillation is a promising direction for multi-teacher post-training and motivate future work on multi-task distillation and scalable multi-teacher training systems.

9 Team Contributions

- **Juntong**: responsible for all aspects of the project, including problem setup, method design, implementation, running experiments, and writing.

References

AI-MO. 2024. AIME 2024. `https://huggingface.co/datasets/AI-MO/aimo-validation-aime`.

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, Qixin Xu, Weize Chen, et al. 2025. Process reinforcement through implicit rewards. *arXiv preprint arXiv:2502.01456* (2025).

DeepSeek-AI. 2026. DeepSeek-V4: Towards Highly Efficient Million-Token Context Intelligence.

Zhiwei He, Tian Liang, Jiahao Xu, Qiuzhi Liu, Xingyu Chen, Yue Wang, Linfeng Song, Dian Yu, Zhenwen Liang, Wenxuan Wang, et al. 2025. Deepmath-103k: A large-scale, challenging, decontaminated, and verifiable mathematical dataset for advancing reasoning. *arXiv preprint arXiv:2504.11456* (2025).

Kevin Lu and Thinking Machines Lab. 2025. On-Policy Distillation. *Thinking Machines Lab: Connectionism* (2025). doi:`10.64434/tml.20251026` https://thinkingmachines.ai/blog/on-policy-distillation.

OpenCompass. 2025. AIME 2025. `https://huggingface.co/datasets/opencompass/AIME2025`.

Stephane Ross, Geoffrey J. Gordon, and J. Andrew Bagnell. 2011. A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning. arXiv:1011.0686 [cs.LG] `https://arxiv.org/abs/1011.0686`

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2024. HybridFlow: A Flexible and Efficient RLHF Framework. *arXiv preprint arXiv: 2409.19256* (2024).

Core Team, Bangjun Xiao, Bingquan Xia, Bo Yang, Bofei Gao, Bowen Shen, Chen Zhang, Chenhong He, Chiheng Lou, Fuli Luo, Gang Wang, Gang Xie, Hailin Zhang, Hanglong Lv, Hanyu Li, Heyu Chen, Hongshen Xu, Houbin Zhang, Huaqiu Liu, Jiangshan Duo, Jianyu Wei, Jiebao Xiao, Jinhao Dong, Jun Shi, Junhao Hu, Kainan Bao, Kang Zhou, Lei Li, Liang Zhao, Linghao Zhang, Peidian Li, Qianli Chen, Shaohui Liu, Shihua Yu, Shijie Cao, Shimao Chen, Shouqiu Yu, Shuo Liu, Tianling Zhou, Weijiang Su, Weikun Wang, Wenhan Ma, Xiangwei Deng, Bohan Mao, Bowen Ye, Can Cai, Chenghua Wang, Chengxuan Zhu, Chong Ma, Chun Chen, Chunan Li, Dawei Zhu, Deshan Xiao, Dong Zhang, Duo Zhang, Fangyue Liu, Feiyu Yang, Fengyuan Shi, Guoan Wang, Hao Tian, Hao Wu, Heng Qu, Hongfei Yi, Hongxu An, Hongyi Guan, Xing Zhang, Yifan Song, Yihan Yan, Yihao Zhao, Yingchun Lai, Yizhao Gao, Yu Cheng, Yuanyuan Tian, Yudong Wang, Zhen Tang, Zhengju Tang, Zhengtao Wen, Zhichao Song, Zhixian Zheng, Zihan Jiang, Jian Wen, Jiarui Sun, Jiawei Li, Jinlong Xue, Jun Xia, Kai Fang, Menghang Zhu, Nuo Chen, Qian Tu, Qihao Zhang, Qiying Wang, Rang Li, Rui Ma, Shaolei Zhang, Shengfan Wang, Shicheng Li, Shuhao Gu, Shuhuai Ren, Sirui Deng, Tao Guo, Tianyang Lu, Weiji Zhuang, Weikang Zhang, Weimin Xiong, Wenshan Huang, Wenyu Yang, Xin Zhang, Xing Yong, Xu Wang, Xueyang Xie, Yilin Jiang, Yixin Yang, Yongzhe He, Yu Tu, Yuanliang Dong, Yuchen Liu, Yue Ma, Yue Yu, Yuxing Xiang, Zhaojun Huang, Zhenru Lin, Zhipeng Xu, Zhiyang Chen, Zhonghua Deng, Zihan Zhang, and Zihao Yue. 2026. MiMo-V2-Flash Technical Report. arXiv:2601.02780 [cs.CL] https://arxiv.org/abs/2601.02780

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. 2025a. Qwen3 technical report. *arXiv preprint arXiv:2505.09388* (2025).

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. 2025b. Qwen3 Technical Report. arXiv:2505.09388 [cs.CL] https://arxiv.org/abs/2505.09388

Chuanguang Yang, Xinqiang Yu, Han Yang, Zhulin An, Chengqing Yu, Libo Huang, and Yongjun Xu. 2025c. Multi-Teacher Knowledge Distillation with Reinforcement Learning for Visual Recognition. arXiv:2502.18510 [cs.CV] https://arxiv.org/abs/2502.18510

Wenkai Yang, Weijie Liu, Ruobing Xie, Kai Yang, Saiyong Yang, and Yankai Lin. 2026. Learning beyond Teacher: Generalized On-Policy Distillation with Reward Extrapolation. arXiv:2602.12125 [cs.LG] https://arxiv.org/abs/2602.12125

Fei Yuan, Linjun Shou, Jian Pei, Wutao Lin, Ming Gong, Yan Fu, and Daxin Jiang. 2020. Reinforced Multi-Teacher Selection for Knowledge Distillation. arXiv:2012.06048 [cs.CL] https://arxiv.org/abs/2012.06048

A Qualitative Examples

Coding Anti-Expert Response:

We are given a word problem involving Aya's walking speed and the time she spends walking and in the coffee shop.
Step 1: Define known quantities and variables

- The total distance of the walk is 9 kilometers.
- Let s be Aya's walking speed in km/h.
- Let t be the number of minutes spent in the coffee shop.

In the first scenario, Aya walks at s km/h and the total time (walking plus coffee shop) is 4 hours. In the second scenario, she walks at $s + 2$ km/h and the total time is 2.4 hours.
...

G-OPD Response:

We are given a problem involving assigning flavors to 9 players, with constraints on the number of players choosing each flavor.
Problem Summary
There are 9 players, each assigned a single scoop cone of chocolate, vanilla, or strawberry.
The key steps are:

1. Find all possible triples (c, v, s).
2. Compute $9!/(c!v!s!)$ for each triple.
3. Sum the results.

...

TC-OPD Response:

We are given a problem involving 9 players who each choose one of three flavors: chocolate, vanilla, or strawberry.
The plan is:

1. Find all possible triples (c, v, s) where $c > v > s$.
2. Compute the number of assignments for each triple.
3. Sum all valid assignments.

Let me work on Step 1: finding the possible triples (c, v, s)
...