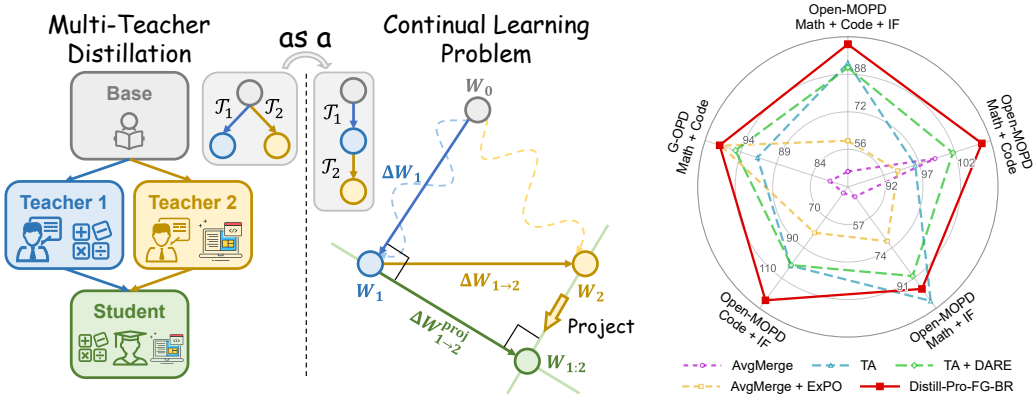
TRAINING-FREE MULTI-TEACHER DISTILLATION AS A CONTINUAL LEARNING PROBLEM

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-teacher distillation (MTD) has achieved notable success in integrating diverse capabilities into a single model; however, existing approaches suffer from immense data and computational costs. While model merging offers a training-free alternative, existing techniques largely rely on empirical heuristics and lack a rigorous theoretical foundation. In this paper, we novelly formulate MTD as a continual learning problem. Specifically, we conceptualize independent teachers fine-tuned from a shared base model as having been trained sequentially, where acquiring a "new" task induces catastrophic forgetting on "previous" ones. Consequently, we propose **Distill**-by-**Pro**jection (Distill-Pro), a post-hoc parameter rectification technique that projects and corrects model updates to eliminate components that interfere with previous task performance. Crucially, Distill-Pro remains entirely data-free and training-free, while being grounded in a theoretical foundation of first-order performance invariance. Extensive evaluations on two large language model multi-teacher distillation benchmarks—encompassing two model architectures and three diverse tasks—demonstrate that our method consistently outperforms or matches state-of-the-art model merging and on-policy distillation baselines.



(a) Schematic diagram (b) Comparison of recovery rates

Figure 1: **Overview of Distill-Pro**. **Left:** Multi-Teacher Distillation seeks to integrate knowledge from independent teachers ($\boldsymbol{W}_1, \boldsymbol{W}_2$) fine-tuned from a shared base model $\boldsymbol{W}_0$. We conceptually reframe this parallel process into a sequential training scheme ($\mathcal{T}_1 \to \mathcal{T}_2$). Distill-Pro rectifies the sequential update $\Delta\boldsymbol{W}_{1\to2}$ via orthogonal projection relative to $\Delta\boldsymbol{W}_1$. **Right:** Recovery Rates (%) comparison across five task combinations on two benchmarks. Our proposed method (**Distill-Pro-FG-BR**, solid red line) achieves superior performance in the vast majority of scenarios.

1 INTRODUCTION

Reinforcement learning (RL) fine-tuning of large language models (LLMs) has achieved remarkable success across a wide range of specialized tasks, including mathematical reasoning (Guo et al.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

(2025)), code generation (Jain et al. (2025b)), and instruction following (Ouyang et al. (2022)). However, real-world applications demand a versatile, unified model that simultaneously encapsulates these diverse capabilities, rather than relying on a collection of narrow experts. A naive approach to achieving this is mixing data from multiple tasks for joint training (Yang et al. (2025)). Unfortunately, this often suffers from inter-task interference, leading to the notorious "seesaw effect" where performance gains on one task come at the expense of another (Standley et al. (2020)). Consequently, jointly trained models often fail to match the performance of domain-specific experts.

To address this, Multi-Teacher Distillation (MTD) has attracted increasing attention. MTD aims to integrate multiple expert models—independently fine-tuned from the same base model across different domains—into a single student model. Typically, this is achieved by extending on-policy distillation (OPD) (Agarwal et al. (2024)) to multi-teacher settings. OPD encourages the student to match the teachers' logit distributions over its own generated trajectories and has proven advantageous as it mitigates exposure bias. Based on OPD, MOPD methods (Ma et al. (2026); Yang et al. (2026); Gao et al. (2026)) can successfully distill the knowledge of multiple teachers into a single student model.

Despite the effectiveness of MOPD methods, these training-based distillation methods are heavily bottlenecked by their reliance on extensive domain data and immense computational resources. In contrast, model merging (Wortsman et al. (2022); Ilharco et al. (2023)) offers a promising training-free alternative, integrating multiple teacher models directly within the weight space to produce a unified model at minimal cost. Nevertheless, existing model merging techniques largely rely on empirical heuristics and lack a rigorous theoretical foundation, often resulting in unstable or suboptimal performance across downstream tasks.

To overcome these limitations, we novelly reformulate the MTD problem through the lens of Continual Learning (CL) (Wang et al. (2024)). Specifically, we conceptualize the independent teacher models as having been trained sequentially on their respective tasks, where learning a "new" task inherently induces catastrophic forgetting on "previous" ones, as illustrated in Fig. 1a. Inspired by prior efforts that mitigate catastrophic forgetting via post-hoc parameter rectification, we propose **Distill**-by-**Pro**jection (**Distill-Pro**), a novel training-free MTD approach. Distill-Pro utilizes the parameter updates ($\Delta \boldsymbol{W}$) of preceding tasks as proxies for their primary gradient directions. By projecting the parameter updates of new tasks, it explicitly neutralizes the components that interfere with prior knowledge before combining them into the final integrated model. Crucially, Distill-Pro remains entirely data-free and training-free, while being grounded in a rigorous theoretical foundation of first-order performance invariance.

Our main contributions are summarized as follows:

- We introduce a novel perspective on MTD, reformulating the integration of multi-domain knowledge as a continual learning problem. This formulation transforms MTD into a process of mitigating catastrophic forgetting via post-hoc parameter update projection.
- Based on this perspective, we propose Distill-Pro, an entirely data-free and training-free method that merges multiple teachers with minimal computational overhead, uniquely supported by the theoretical foundation of first-order performance invariance.
- Extensive evaluations on two LLM MTD benchmarks—encompassing two distinct architectures and three diverse tasks—demonstrate that Distill-Pro consistently outperforms or matches state-of-the-art model merging and MOPD baselines.

2 Related Works

Multi-teacher Distillation Traditional Knowledge Distillation (Hinton et al. (2015)) aligns the student with the teacher's logits over teacher-generated trajectories. Recently, OPD (Agarwal et al. (2024); Gu et al. (2023)) has emerged as a more effective paradigm by aligning over the student's own generated trajectories, thereby mitigating distribution shifts. Ma et al. (2026) and Xiao et al. (2026) first extend OPD to the multi-teacher setting, and propose MOPD. Open-MOPD (Gao et al. (2026)) addresses cross-domain optimization imbalance through token-share balancing, gap-aware dynamic budget allocation, and online refresh of student-dependent rewards.

Model Merging Model merging integrates multiple independently-finetuned models directly within the weight space. Foundational methods like Model Soups (Wortsman et al. (2022)) average the parameters of each teacher, while Task Arithmetic (TA) (Ilharco et al. (2023)) linearly combines task-specific parameter updates. To alleviate parameter conflicts during integration, advanced techniques such as TIES-Merging (Yadav et al. (2023)) and DARE (Yu et al. (2024)) employ sparsification and consensus-based masking. Despite their computational efficiency, their effectiveness largely depends on empirically designed merging rules and lacks a rigorous theoretical foundation against inter-task interference, often leading to unstable performance.

Mitigating Catastrophic Forgetting CL focuses on preserving previously acquired knowledge when a model sequentially learns new tasks. Typical anti-forgetting mechanisms fall into three broad categories: regularization-based (Kirkpatrick et al. (2017); Liu et al. (2018); Wang et al. (2021)), optimization-based (Lopez-Paz & Ranzato (2017); Riemer et al. (2019); Farajtabar et al. (2020)), and architecture-based methods (Aljundi et al. (2017); Mallya et al. (2018); Serra et al. (2018)). However, these traditional methods require intervention during the dynamic training phase of new tasks, rendering them inapplicable to the MTD setting where off-the-shelf teacher models are already provided. Recent work (Zheng et al. (2026)) introduces post-hoc parameter rectification to recover previous task performance. However, it reintroduces computational overheads by relying on historical task data and gradients.

3 PRELIMINARY

Let $\{\mathcal{T}_i\}_{i=1}^T$ denote T target tasks we aim for the model to learn, and let $\{\mathcal{L}_i\}_{i=1}^T$ be their corresponding loss functions. Consider a neural network consisting of L submodules. Suppose we are provided with T teacher models, each independently fine-tuned on one of these tasks from a shared base model parameterized by $\{\boldsymbol{W}_0^{(l)}\}_{l=1}^L$, where $\boldsymbol{W}_0^{(l)}$ denotes the parameters of the l-th submodule. With a slight abuse of notation, we use $\boldsymbol{W}_0^{(l)}$ to represent the parameters regardless of whether the submodule is characterized by a matrix or a vector. Let $\{\boldsymbol{W}_i^{(l)}\}_{l=1}^L$ denote the parameters of the i-th teacher model. Consequently, each teacher corresponds to a task-specific parameter update $\{\Delta\boldsymbol{W}_i^{(l)} \triangleq \boldsymbol{W}_i^{(l)} - \boldsymbol{W}_0^{(l)}\}_{l=1}^L$. For notational simplicity, we focus on a single specific submodule in the remainder of the paper and omit the superscript (l). To reformulate MTD as a CL problem, we conceptually assume that these teacher models are obtained by training on the T tasks sequentially in the order of $\mathcal{T}_1 \to \mathcal{T}_2 \to \cdots \to \mathcal{T}_T$. Under this sequential assumption, we define $\Delta\boldsymbol{W}_{i-1\to i} \triangleq \boldsymbol{W}_i - \boldsymbol{W}_{i-1}$ for $i = 1, \ldots, T$ as the conceptual parameter update induced by learning task $\mathcal{T}_i$. It naturally follows that $\boldsymbol{W}_i = \boldsymbol{W}_0 + \sum_{j=1}^i \Delta\boldsymbol{W}_{j-1\to j}$.

4 METHODS

4.1 PARAMETER UPDATE PROJECTION

Let us first consider a two-task scenario. After sequentially learning task $\mathcal{T}_2$, the introduced parameter update $\Delta\boldsymbol{W}_{1\to 2}$ makes the model an expert on $\mathcal{T}_2$, but leads to catastrophic forgetting on $\mathcal{T}_1$. Formally, this implies that

$$\Delta\mathcal{L}_1 \triangleq \mathcal{L}_1(\boldsymbol{W}_2) - \mathcal{L}_1(\boldsymbol{W}_1) = \mathcal{L}_1(\boldsymbol{W}_1 + \Delta\boldsymbol{W}_{1\to 2}) - \mathcal{L}_1(\boldsymbol{W}_1) > 0. \tag{1}$$

Performing a first-order Taylor expansion on $\mathcal{L}_1$ around $\boldsymbol{W}_1$, we obtain

$$\Delta\mathcal{L}_1 = \langle \nabla_{\boldsymbol{W}}\mathcal{L}_1(\boldsymbol{W}_1), \Delta\boldsymbol{W}_{1\to 2}\rangle + o(\|\Delta\boldsymbol{W}_{1\to 2}\|) \triangleq \langle \boldsymbol{G}_1, \Delta\boldsymbol{W}_{1\to 2}\rangle + o(\|\Delta\boldsymbol{W}_{1\to 2}\|), \tag{2}$$

where $\langle\cdot,\cdot\rangle$ denotes the inner product and $\|\cdot\|$ is the norm. In cases where $\boldsymbol{W}$ is a matrix, $\langle\cdot,\cdot\rangle$ represents the Frobenius inner product, given by $\langle \boldsymbol{A}, \boldsymbol{B}\rangle_F = \mathrm{Tr}(\boldsymbol{A}^\top\boldsymbol{B})$. Consequently, if the new parameter update $\Delta\boldsymbol{W}_{1\to 2}$ is orthogonal to the gradient $\boldsymbol{G}_1$ (i.e., $\langle \boldsymbol{G}_1, \Delta\boldsymbol{W}_{1\to 2}\rangle = 0$), it follows that the change in $\mathcal{L}_1$ induced by $\Delta\boldsymbol{W}_{1\to 2}$ is approximately zero up to the first order:

$$\Delta\mathcal{L}_1 = \langle \boldsymbol{G}_1, \Delta\boldsymbol{W}_{1\to 2}\rangle + o(\|\Delta\boldsymbol{W}_{1\to 2}\|_2) = o(\|\Delta\boldsymbol{W}_{1\to 2}\|_2) \approx 0. \tag{3}$$

A recent study (Zheng et al. (2026)) has demonstrated that by projecting parameter updates onto the Jacobian null space of a prior task, a model can recover its forgotten capabilities. Inspired by this

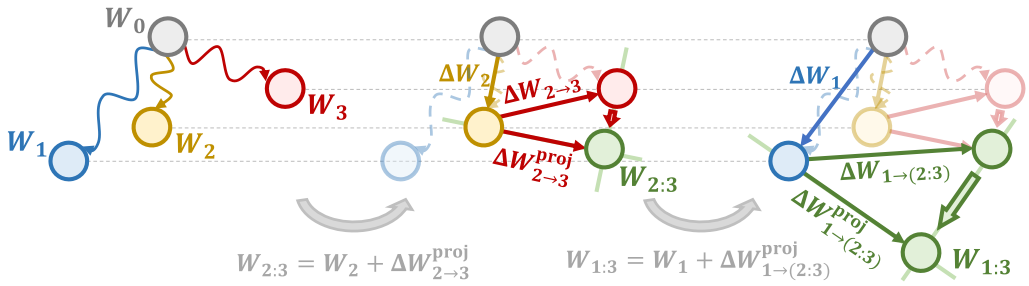


Figure 2: Illustration of the multi-task projection process.

insight, we propose projecting $\Delta \boldsymbol{W}_{1\to2}$ to rectify it, ensuring its orthogonality with $\boldsymbol{G}_1$:

$$\Delta \boldsymbol{W}^{\text{proj}}_{1\to2} = \Delta \boldsymbol{W}_{1\to2} - \frac{\langle \boldsymbol{G}_1, \Delta \boldsymbol{W}_{1\to2} \rangle}{\langle \boldsymbol{G}_1, \boldsymbol{G}_1 \rangle} \boldsymbol{G}_1. \tag{4}$$

The rectified update $\Delta \boldsymbol{W}^{\text{proj}}_{1\to2}$ is the exact solution to the following first-order constrained problem:

$$\begin{aligned} \min_{\Delta \boldsymbol{W}} \quad & \frac{1}{2} \|\Delta \boldsymbol{W} - \Delta \boldsymbol{W}_{1\to2}\|_F^2 \\ \text{s.t.} \quad & \langle \boldsymbol{G}_1, \Delta \boldsymbol{W} \rangle = 0. \end{aligned} \tag{5}$$

Geometrically, this yields the parameter update closest to the original $\Delta \boldsymbol{W}_{1\to2}$ that simultaneously maintains orthogonality with $\boldsymbol{G}_1$.

However, obtaining the exact $\boldsymbol{G}_1$ requires performing forward and backward passes across the network using $\mathcal{T}_1$ data, which reintroduces significant computational overhead. Therefore, it is desirable to find an efficient proxy for $\boldsymbol{G}_1$. Previous studies on optimization dynamics have established the parameter update $\Delta \boldsymbol{W}$ as a critical direction in the parameter space. For instance, Nichol et al. (2018) demonstrated that this direction encourages the gradients of different mini-batches to become more aligned. Similarly, Gur-Ari et al. (2018) found that gradients quickly converge to a low-dimensional subspace spanned by the top k eigenvectors of the Hessian. Consequently, the direction of $\Delta \boldsymbol{W}$ serves as an effective representation of the primary negative gradient direction, or equivalently, the high-curvature subspace of the loss landscape. Zhou et al. (2025) has also investigated this from a theoretical perspective. Building on these insights, we propose adopting $\Delta \boldsymbol{W}_1$ as a proxy for $\boldsymbol{G}_1$. The projection process then becomes

$$\Delta \boldsymbol{W}^{\text{proj}}_{1\to2} = \Delta \boldsymbol{W}_{1\to2} - \frac{\langle \Delta \boldsymbol{W}_1, \Delta \boldsymbol{W}_{1\to2} \rangle}{\langle \Delta \boldsymbol{W}_1, \Delta \boldsymbol{W}_1 \rangle} \Delta \boldsymbol{W}_1. \tag{6}$$

After rectifying the parameter update, the integrated model is given by:

$$\boldsymbol{W}_{1:2} = \boldsymbol{W}_0 + \Delta \boldsymbol{W}_1 + \Delta \boldsymbol{W}^{\text{proj}}_{1\to2} = \boldsymbol{W}_1 + \Delta \boldsymbol{W}^{\text{proj}}_{1\to2}. \tag{7}$$

To generalize this to T tasks, we integrate the teacher models in reverse sequential order. As illustrated in Fig. 2, we first perform projection rectification on $\Delta \boldsymbol{W}_{T-1\to T}$ with respect to $\Delta \boldsymbol{W}_{T-1}$, yielding $\boldsymbol{W}_{T-1:T} = \boldsymbol{W}_{T-1} + \Delta \boldsymbol{W}^{\text{proj}}_{T-1\to T}$. Next, we project $\Delta \boldsymbol{W}_{T-2\to(T-1:T)}$ with respect to $\Delta \boldsymbol{W}_{T-2}$, obtaining $\boldsymbol{W}_{T-2:T} = \boldsymbol{W}_{T-2} + \Delta \boldsymbol{W}^{\text{proj}}_{T-2\to(T-1:T)}$. This recursive process continues until we obtain the final integrated model $\boldsymbol{W}_{1:T} = \boldsymbol{W}_1 + \Delta \boldsymbol{W}^{\text{proj}}_{1\to(2:T)}$.

4.2 More Fine-Grained Projection

The majority of parameters in the model belong to the weight matrices of linear layers, where $\boldsymbol{W} \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$. Let ${\boldsymbol{w}^{(i)}}^\top$, $i = 1, \ldots, d_{\text{out}}$ denote the i-th row of $\boldsymbol{W}$. Then, the inner product between

$\Delta \boldsymbol{W}_1$ and $\Delta \boldsymbol{W}_{1\to 2}$ can be decomposed as:

$$\begin{aligned}\langle \Delta \boldsymbol{W}_1, \Delta \boldsymbol{W}_{1\to 2}\rangle &= \mathrm{Tr}\big(\Delta \boldsymbol{W}_1^\top \Delta \boldsymbol{W}_{1\to 2}\big)\\ &= \mathrm{Tr}\big(\Delta \boldsymbol{W}_{1\to 2} \Delta \boldsymbol{W}_1^\top\big)\\ &= \mathrm{Tr}\left(\left[\Delta \boldsymbol{w}_{1\to 2}^{(1)}, \ldots, \Delta \boldsymbol{w}_{1\to 2}^{(d_{\text{out}})}\right]^\top \left[\Delta \boldsymbol{w}_1^{(1)}, \ldots, \Delta \boldsymbol{w}_1^{(d_{\text{out}})}\right]\right)\\ &= \sum_{i=1}^{d_{\text{out}}} {\Delta \boldsymbol{w}_{1\to 2}^{(i)}}^\top \Delta \boldsymbol{w}_1^{(i)}\\ &= \sum_{i=1}^{d_{\text{out}}} \left\langle \Delta \boldsymbol{w}_1^{(i)}, \Delta \boldsymbol{w}_{1\to 2}^{(i)} \right\rangle.\end{aligned} \tag{8}$$

Hence, we can perform a more fine-grained projection by making each row of $\Delta \boldsymbol{W}_{1\to 2}$ orthogonal to the corresponding row of $\Delta \boldsymbol{W}_1$:

$$\Delta \boldsymbol{w}_{1\to 2}^{(i),\text{proj}} = \Delta \boldsymbol{w}_{1\to 2}^{(i)} - \frac{\left\langle \Delta \boldsymbol{w}_1^{(i)}, \Delta \boldsymbol{w}_{1\to 2}^{(i)} \right\rangle}{\left\langle \Delta \boldsymbol{w}_1^{(i)}, \Delta \boldsymbol{w}_1^{(i)} \right\rangle} \Delta \boldsymbol{w}_1^{(i)}. \tag{9}$$

We term this variant as **Fine-Grained Distill-Pro** (Distill-Pro-FG).

4.3 Relation with Task Arithmetic

Interestingly, noting that $\Delta \boldsymbol{W}_{1\to 2} = (\boldsymbol{W}_2 - \boldsymbol{W}_0) - (\boldsymbol{W}_1 - \boldsymbol{W}_0) = \Delta \boldsymbol{W}_2 - \Delta \boldsymbol{W}_1$, we have

$$\begin{aligned}\Delta \boldsymbol{W}_{1\to 2}^{\text{proj}} &= \Delta \boldsymbol{W}_{1\to 2} - \frac{\langle \Delta \boldsymbol{W}_1, \Delta \boldsymbol{W}_{1\to 2}\rangle}{\langle \Delta \boldsymbol{W}_1, \Delta \boldsymbol{W}_1\rangle} \Delta \boldsymbol{W}_1\\ &= \Delta \boldsymbol{W}_2 - \Delta \boldsymbol{W}_1 - \frac{\langle \Delta \boldsymbol{W}_1, \Delta \boldsymbol{W}_2 - \Delta \boldsymbol{W}_1\rangle}{\langle \Delta \boldsymbol{W}_1, \Delta \boldsymbol{W}_1\rangle} \Delta \boldsymbol{W}_1\\ &= \Delta \boldsymbol{W}_2 - \Delta \boldsymbol{W}_1 - \frac{\langle \Delta \boldsymbol{W}_1, \Delta \boldsymbol{W}_2\rangle}{\langle \Delta \boldsymbol{W}_1, \Delta \boldsymbol{W}_1\rangle} \Delta \boldsymbol{W}_1 + \Delta \boldsymbol{W}_1\\ &= \Delta \boldsymbol{W}_2^{\text{proj}}.\end{aligned} \tag{10}$$

This equivalence reveals that the integrated model $\boldsymbol{W}_{1:2}$ can also be constructed by projecting the base-referenced parameter update $\Delta \boldsymbol{W}_2$ with respect to $\Delta \boldsymbol{W}_1$, yielding $\boldsymbol{W}_{1:2} = \boldsymbol{W}_0 + \Delta \boldsymbol{W}_1 + \Delta \boldsymbol{W}_2^{\text{proj}}$. We term this variant **Base-Referenced Distill-Pro** (Distill-Pro-BR). Without the projection step, this formulation reduces to $\boldsymbol{W}_{1:2}^{\text{TA}} = \boldsymbol{W}_0 + \sum_{i=1}^{2} \Delta \boldsymbol{W}_i$, which recovers the TA method (Ilharco et al. (2023)). Hence, TA can be viewed as an ablated baseline of Distill-Pro-BR.

5 Experiments

5.1 Experimental settings

Benchmarks To ensure a comprehensive and reliable evaluation, we adopt two benchmarks from Open-MOPD (Gao et al. (2026)) and G-OPD (Yang et al. (2026)), covering two distinct model families across three domains. Specifically, the Open-MOPD benchmark constructs its base model by applying mixed-domain supervised fine-tuning to SmolLM3-3B-Base, involving three tasks: Math, Code, and Instruction Following (IF). Across these three tasks, we evaluate four combinations: Math + Code, Math + IF, Code + IF, and Math + Code + IF. The G-OPD benchmark employs Qwen3-4B-Non-Thinking (Yang et al. (2025)) as the base model and focuses on two tasks: Math and Code. For more details regarding the training of teacher models, we refer readers to (Gao et al. (2026)) and (Yang et al. (2026)).

Evaluations We strictly follow the original evaluation protocols for both benchmarks, adhering to their designated datasets and hyperparameter configurations. Specifically, on the Open-MOPD

benchmark, the Math task is evaluated on AIME24 (Zhang & Math-AI (2024)) and AIME25 (Zhang & Math-AI (2025)); the Code task on LiveCodeBench (v5 and v6) (Jain et al. (2025a)); and the IF task on IFEval (Zhou et al. (2023)) and $\text{IFBench}_{\text{test}}$ (Pyatkin et al. (2025)). For generation, we set top-p to 0.95 and disable top-k truncation ($k = -1$). We sample 64 responses per math problem at a temperature of 0.6, 10 responses per code problem at a temperature of 1.0, and 1 response per IF prompt at a temperature of 1.0. On the G-OPD benchmark, the Math task is evaluated on AIME24, AIME25, HMMT25 (Feb.), and HMMT25 (Nov.) (Balunovic et al. (2025)). The Code task is evaluated on HumanEval+, MBPP+ (Liu et al. (2023)), and LiveCodeBench (v6 only). For generation, we set both the temperature and top-p to 1.0, sampling 32 responses per math problem and 4 responses per code problem.

Metrics Following prior work, we report the average accuracy for each benchmark. We define the *Route Expert* as an oracle reference, constructed by aggregating the in-domain performance of the corresponding RL domain experts. We also report *Total*, which is the average of domain-level scores (where each domain score is averaged across its constituent datasets), and the *Recovery Rate* (Rec. Rate), defined as $\frac{\text{Total}_{\text{method}} - \text{Total}_{\text{Base}}}{\text{Total}_{\text{Route Expert}} - \text{Total}_{\text{Base}}}$. It measures the proportion of the performance gap recovered relative to the Route Expert oracle.

Baselines We compare our method against two categories of baselines. Detailed implementation configurations for all baselines are provided in Appendix A.

1. **Model Merging Methods:** (1) **Average Merging (AvgMerge)** (Wortsman et al. (2022)) directly averages parameters across all expert models; (2) **ExPO** (Zheng et al. (2025)) extrapolates the averaged model parameters away from the base model; (3) **TA** (Ilharco et al. (2023)) linearly combines task-specific parameter updates; and (4) **DARE** (Yu et al. (2024)) augments TA by mitigating parameter conflicts during integration.

2. **MOPD Methods:** (5) **Naive MOPD** (Ma et al. (2026)) distills multiple teacher models into a single student via OPD; (6) **ExOPD** (Yang et al. (2026)) extrapolates rewards with a scaling factor greater than 1; and (7) **Open-MOPD** (Gao et al. (2026)) addresses cross-domain optimization imbalance during distillation.

5.2 Main results

Tables 1 and 2 list the main results on the Open-MOPD and G-OPD benchmarks, respectively.

On the Open-MOPD benchmark, the Distill-Pro family achieves the highest overall performance across all four task combinations, consistently outperforming both Model Merging and MOPD baselines. Specifically, Distill-Pro and its variants consistently achieve strong Recovery Rates on Open-MOPD, often exceeding 100% and surpassing individual teacher models across various task combinations. Notably, the advantage of Distill-Pro over baseline methods becomes more pronounced in the three-task setting (Task Combination 4).

On the G-OPD benchmark, the Distill-Pro family yields performance highly comparable to MOPD baselines. Its variants reach a Total score up to 52.64, closely approaching the 53.81 and 53.88 achieved by Naive MOPD and ExOPD, respectively. Crucially, unlike these training-based approaches, Distill-Pro is completely training-free and constructs the integrated model via lightweight parameter operations in just a few minutes. It thus offers a significantly more cost-effective alternative for unifying the capabilities of multiple domain teachers into a single model.

To assess performance stability across task combinations, Fig. 1b visualizes the aggregated metrics across five task combinations using a radar plot. As shown in Fig. 1b, Distill-Pro-FG-BR exhibits enhanced performance stability across combinations. This confirms that our CL formulation, anchored by the theoretical foundation of first-order performance invariance, renders training-free MTD more reliable.

Table 1: Results for different task combinations under the Open-MOPD benchmark. For each column, the **best** and second-best results are shown in **bold** and underlined, respectively. In each task combination, the method with the highest recovery rate, is highlighted in light blue.

Methods	Math (avg@64)			Code (avg@10)			IF			Total	Rec. Rate (%)
	$AIME_{24}$	$AIME_{25}$	Avg.	LCBv5	LCBv6	Avg.	IFEval	IFB_{test}	Avg.		
Task Combination 1: Math + Code											
Base	15.16	17.40	16.28	14.85	18.69	16.77	-	-	-	16.52	-
Route Expert	23.02	24.69	23.85	22.63	21.94	22.29	-	-	-	23.07	100
AvgMerge	**24.90**	24.11	24.51	20.78	22.29	21.53	-	-	-	23.02	99.19
AvgMerge + ExPO	24.22	23.02	23.62	**23.41**	20.06	21.74	-	-	-	22.68	93.98
TA	23.85	24.69	24.27	21.44	21.37	21.40	-	-	-	22.84	96.43
TA + DARE	23.49	24.79	24.14	22.34	22.11	22.22	-	-	-	23.18	101.70
Distill-Pro	24.27	**24.84**	**24.56**	22.16	22.00	22.08	-	-	-	23.32	103.76
Distill-Pro-BR	23.80	23.54	23.67	22.10	21.60	21.85	-	-	-	22.76	95.24
Distill-Pro-FG	24.64	24.17	24.40	21.98	21.83	21.90	-	-	-	23.15	101.22
Distill-Pro-FG-BR	24.48	23.80	24.14	23.11	**22.40**	**22.76**	-	-	-	**23.45**	**105.76**
Task Combination 2: Math + IF											
Base	15.16	17.40	16.28	-	-	-	63.40	20.33	41.87	29.07	-
Route Expert	23.02	24.69	23.85	-	-	-	74.49	24.00	49.25	36.55	100
AvgMerge	19.64	21.46	20.55	-	-	-	69.13	19.67	44.40	32.47	45.48
AvgMerge + ExPO	20.78	22.08	21.43	-	-	-	74.49	20.00	47.25	34.34	70.44
TA	23.59	24.32	23.96	-	-	-	75.42	24.00	49.71	36.83	103.79
TA + DARE	24.69	22.40	23.54	-	-	-	73.75	22.33	48.04	35.79	89.87
Distill-Pro	**24.84**	24.27	**24.56**	-	-	-	74.31	21.33	47.82	36.19	95.17
Distill-Pro-BR	23.80	**24.95**	24.38	-	-	-	**75.79**	**26.00**	**50.89**	**37.63**	**114.49**
Distill-Pro-FG	24.17	24.79	24.48	-	-	-	75.23	22.67	48.95	36.71	102.19
Distill-Pro-FG-BR	23.44	24.32	23.88	-	-	-	73.57	24.00	48.78	36.33	97.08
Task Combination 3: Code + IF											
Base	-	-	-	14.85	18.69	16.77	63.40	20.33	41.87	29.32	-
Route Expert	-	-	-	22.63	21.94	22.29	74.49	24.00	49.25	35.77	100
AvgMerge	-	-	-	20.12	20.11	20.12	70.98	20.00	45.49	32.80	54.05
AvgMerge + ExPO	-	-	-	21.26	22.06	21.66	70.98	23.67	47.32	34.49	80.20
TA	-	-	-	23.83	23.03	23.43	73.01	23.67	48.34	35.89	101.83
TA + DARE	-	-	-	23.29	22.40	22.85	72.09	25.67	48.88	35.86	101.47
Distill-Pro	-	-	-	24.55	23.31	23.93	72.27	22.33	47.30	35.62	97.69
Distill-Pro-BR	-	-	-	23.41	**23.60**	23.51	73.57	23.00	48.28	35.90	101.98
Distill-Pro-FG	-	-	-	22.81	22.97	22.89	71.53	23.67	47.60	35.25	91.93
Distill-Pro-FG-BR	-	-	-	**25.21**	22.91	**24.06**	**74.31**	**27.00**	**50.65**	**37.36**	**124.66**
Task Combination 4: Math + Code + IF											
Base	15.16	17.40	16.28	14.85	18.69	16.77	63.40	20.33	41.87	24.97	-
Route Expert	23.02	24.69	23.85	22.63	21.94	22.29	74.49	24.00	49.25	31.80	100
AvgMerge	19.84	21.98	20.91	18.26	19.83	19.05	69.32	19.67	44.49	28.15	46.58
AvgMerge + ExPO	21.93	23.07	22.50	20.60	18.80	19.70	71.53	18.33	44.93	29.04	59.68
TA	23.33	**25.00**	24.17	23.05	22.06	22.56	73.38	21.00	47.19	31.30	92.80
TA + DARE	23.80	23.18	23.49	23.53	22.57	23.05	70.61	23.33	46.97	31.17	90.84
Naive MOPD	20.92	21.60	21.26	17.53	20.99	19.26	68.61	18.67	43.64	28.05	45.17
Open-MOPD	23.28	23.33	23.31	20.00	22.51	21.26	73.38	24.00	48.69	31.09	89.58
Distill-Pro	**24.32**	24.32	24.32	23.71	23.31	23.51	71.35	**26.33**	48.84	32.23	106.29
Distill-Pro-BR	23.59	24.84	24.22	23.17	**23.43**	23.30	72.64	25.67	49.15	32.22	106.28
Distill-Pro-FG	24.27	24.64	**24.45**	**24.49**	22.74	**23.62**	**74.68**	25.33	**50.00**	**32.69**	**113.12**
Distill-Pro-FG-BR	23.96	23.96	23.96	24.13	22.40	23.27	72.64	24.00	48.32	31.85	100.77

Table 2: Results under the G-OPD benchmark. For each column, the **best** and second-best results are shown in **bold** and underlined, respectively.

Methods	Math (avg@32)					Code (avg@4)				Total	Rec. Rate (%)
	$AIME_{24}$	$AIME_{25}$	$HMMT_{25}$ (Feb.)	$HMMT_{25}$ (Nov.)	Avg.	HumanEval+	MBPP+	LCBv6	Avg.		
Task Combination 5: Math + Code											
Base	24.90	21.46	10.94	7.40	16.17	76.98	64.02	18.00	53.00	34.59	-
Route Expert	60.42	54.79	31.77	37.81	46.20	84.60	69.91	26.14	60.22	53.21	100
AvgMerge	51.67	46.46	23.33	31.87	38.33	86.28	69.58	27.71	61.19	49.76	81.49
AvgMerge + ExPO	54.58	52.92	29.48	34.58	42.89	**86.43**	70.97	**29.57**	**62.32**	52.61	96.77
TA	56.77	50.83	30.42	36.88	43.72	81.25	69.51	28.00	59.59	51.66	91.66
TA + DARE	57.92	52.08	29.17	35.73	43.72	82.32	71.03	28.86	60.74	52.23	94.75
Naive MOPD	58.75	**54.90**	**32.29**	37.50	45.86	85.82	**72.16**	27.29	61.75	53.81	103.22
ExOPD	**61.25**	**54.90**	31.25	**38.33**	**46.43**	85.21	70.50	28.29	61.33	**53.88**	**103.63**
Distill-Pro	56.35	51.46	31.04	34.79	43.41	82.77	70.83	28.57	60.73	52.07	93.88
Distill-Pro-BR	57.29	50.52	29.27	36.67	43.44	82.32	70.57	28.43	60.44	51.94	93.18
Distill-Pro-FG	57.50	50.73	30.94	36.25	43.85	82.16	70.04	28.29	60.16	52.01	93.56
Distill-Pro-FG-BR	57.40	52.81	30.94	35.73	44.22	82.47	71.43	29.29	61.06	52.64	96.95

Table 3: Results for different task orders for Task Combination 4: Math + Code + IF under the Open-MOPD benchmark.

Methods	Math (avg@64)			Code (avg@10)			IF			Total	Rec. Rate (%)
	$AIME_{24}$	$AIME_{25}$	Avg.	LCBv5	LCBv6	Avg.	IFEval	IFB_{test}	Avg.		
Base	15.16	17.40	16.28	14.85	18.69	16.77	63.40	20.33	41.87	24.97	-
Route Expert	23.02	24.69	23.85	22.63	21.94	22.29	74.49	24.00	49.25	31.80	100
Task Order 1: Math → IF → Code											
Distill-Pro	24.32	24.32	24.32	23.71	23.31	23.51	71.35	26.33	48.84	32.23	106.29
Distill-Pro-BR	23.59	24.84	24.22	23.17	23.43	23.30	72.64	25.67	49.15	32.22	106.28
Distill-Pro-FG	24.27	24.64	24.45	24.49	22.74	23.62	74.68	25.33	50.00	32.69	113.12
Distill-Pro-FG-BR	23.96	23.96	23.96	24.13	22.40	23.27	72.64	24.00	48.32	31.85	100.77
Task Order 2: Code → IF → Math											
Distill-Pro	23.44	24.06	23.75	24.31	23.89	24.10	71.90	23.00	47.45	31.77	99.57
Distill-Pro-BR	24.32	24.01	24.17	23.29	22.06	22.68	70.61	27.00	48.80	31.88	101.26
Distill-Pro-FG	23.91	24.53	24.22	23.71	22.91	23.31	75.05	21.67	48.36	31.96	102.44
Distill-Pro-FG-BR	25.21	24.43	24.82	23.95	22.97	23.46	72.83	25.67	49.25	32.51	110.44
Task Order 3: IF → Math → Code											
Distill-Pro	24.27	24.48	24.38	22.81	22.34	22.58	73.57	25.67	49.62	32.19	105.77
Distill-Pro-BR	23.59	24.32	23.96	23.89	22.57	23.23	72.46	24.67	48.56	31.92	101.78
Distill-Pro-FG	24.32	24.90	24.61	25.21	22.91	24.06	73.01	22.33	47.67	32.11	104.67
Distill-Pro-FG-BR	23.70	23.65	23.67	23.17	22.00	22.59	74.49	23.33	48.91	31.72	98.94

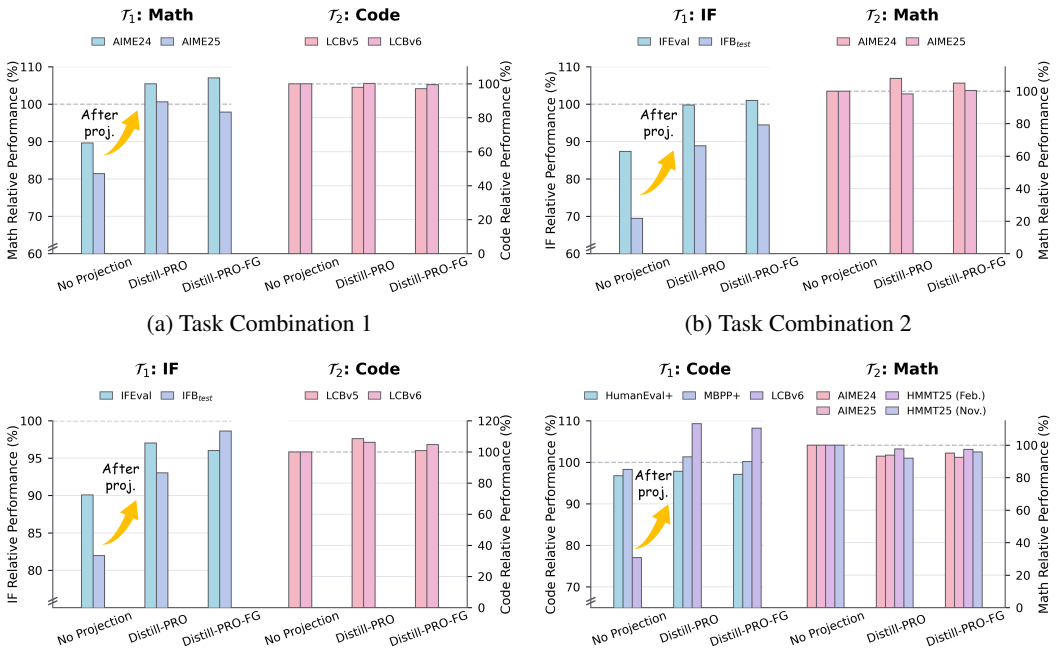
5.3 Ablations

5.3.1 Impact of Task Order

Since Distill-Pro conceptualizes MTD as a CL problem, the sequential order in which task-specific updates are integrated could hypothetically influence final model performance. To investigate this, we evaluate three distinct integration permutations for the Math + Code + IF task combination. As reported in Tab. 3, Distill-Pro and its variants achieve highly consistent overall performance across all three orders. Take vanilla Distill-Pro as an instance. It yields Total scores of 32.23, 31.77, and 32.19 for Math → IF → Code, Code → IF → Math, and IF → Math → Code, respectively. Notably, the performance margin between the highest and lowest Total scores is a mere 0.46 points. This order robustness extends across the entire Distill-Pro family. Taken together, these findings demonstrate that Distill-Pro and its variants are highly resilient to task integration order, validating the soundness and practical utility of our method. The task orders used to obtain the results in Tables 1 and 2 are provided in Appendix A.

5.3.2 Effect of Projection

In two-task scenarios, without the projection step, Distill-Pro-BR reduces to TA; we refer readers to Tables 1 and 2 for detailed performance comparisons. For vanilla Distill-Pro, removing the projection process directly collapses the integrated model into the parameters of the $\mathcal{T}_2$ teacher, since $\boldsymbol{W}_2 = \boldsymbol{W}_1 + \Delta\boldsymbol{W}_{1\to 2}$. Fig. 3 compares the performance of the unrectified $\mathcal{T}_2$ teacher against our projected models on both $\mathcal{T}_1$ and $\mathcal{T}_2$. The results demonstrate that by rectifying components in $\Delta\boldsymbol{W}_{1\to 2}$ that interfere with $\Delta\boldsymbol{W}_1$, the performance on $\mathcal{T}_1$ increases significantly to match the level of the $\mathcal{T}_1$ teacher, while performance on $\mathcal{T}_2$ remains close to the $\mathcal{T}_2$ teacher. We provide the complete results in Appendix. B.



(a) Task Combination 1

(b) Task Combination 2

(c) Task Combination 3

(d) Task Combination 5

Figure 3: Effect of parameter projection across different task combinations. All performance scores are normalized relative to the corresponding standalone teacher models.

6 Conclusion

In this work, we address the challenge of efficiently consolidating multi-domain expertise into a single model without incurring training overhead or data reliance. By establishing a novel connection between multi-teacher distillation and continual learning, we reframe MTD as a sequential parameter rectification process designed to mitigate catastrophic forgetting. Building upon this perspective, we introduce Distill-Pro, a lightweight, data-free, and training-free framework that employs parameter update projections to eliminate cross-task interference, grounded in the principle of first-order performance invariance. Extensive experiments across diverse benchmarks, architectures, and domains confirm that Distill-Pro matches or surpasses both state-of-the-art training-based distillation methods and existing model merging baselines.

AI USE STATEMENT

In this work, we used generative AI tools for implementing methods. We have not used generative AI tools for help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, design or provide feedback on research methodology or experiments, assist with translation, support qualitative and thematic data analysis, interpret results. And generate synthetic data sets, clean and reformat dataset are not applicable to this work. Additionally, we used generative AI tools for summarizing or analyzing existing literature, editing a research paper to improve readability, proposing a title or keywords for a research paper. We have reviewed all AI-assisted work. For example, we verified that all statements summarized by the LLM accurately correspond to the original text. Furthermore, all language edits made by the LLM were subject to further manual review and revision by the authors. LLM-generated code was verified and tested for correctness by 2 authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We present the evaluation protocols, hyperparameters, and baseline implementations in Secs. 5.1 and A. Additionally, Sec. 5.3.1 outlines the task orders used for the main results. The code of Distill-Pro is also provided in the supplementary materials.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=3zKtaqxLhW.

Rahaf Aljundi, Punarjay Chakravarty, and Tinne Tuytelaars. Expert gate: Lifelong learning with a network of experts. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3366–3375, 2017.

Mislav Balunovic, Jasper Dekoninck, Ivo Petrov, Nikola Jovanovic, and Martin T. Vechev. Matharena: Evaluating llms on uncontaminated math competitions. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/1d27c01ebd3e3aebe226b44fc970d803-Abstract-Datasets_and_Benchmarks_Track.html.

Mehrdad Farajtabar, Navid Azizan, Alex Mott, and Ang Li. Orthogonal gradient descent for continual learning. In *International conference on artificial intelligence and statistics*, pp. 3762–3773. PMLR, 2020.

Huan-ang Gao, Haohan Chi, Yong Yan, Shiyuan Feng, Hanlin Wu, Zheng Jiang, Bingxiang He, Wei-Ying Ma, Ya-Qin Zhang, and Hao Zhou. Open-mopd: Diagnosing and fixing capability imbalance in multi-teacher on-policy distillation. *arXiv preprint arXiv:2608.19098*, 2026.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: On-policy distillation of large language models. *arXiv preprint arXiv:2306.08543*, 2023.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li,

Hao Zhang, Hanwei Xu, Honghui Ding, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jingchang Chen, Jingyang Yuan, Jinhao Tu, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaichao You, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingxu Zhou, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Tao Yun, Tian Pei, Tianyu Sun, Tao Wang, Wangding Zeng, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nat.*, 645(8081):633–638, 2025. doi: 10.1038/S41586-025-09422-Z. URL https://doi.org/10.1038/s41586-025-09422-z.

Guy Gur-Ari, Daniel A. Roberts, and Ethan Dyer. Gradient descent happens in a tiny subspace. *CoRR*, abs/1812.04754, 2018. URL http://arxiv.org/abs/1812.04754.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531, 2015. URL http://arxiv.org/abs/1503.02531.

Gabriel Ilharco, Marco Túlio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=6t0Kwf8-jrj.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. Livecodebench: Holistic and contamination free evaluation of large language models for code. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025a. URL https://openreview.net/forum?id=chfJJYC3iL.

Naman Jain, Jaskirat Singh, Manish Shetty, Liang Zheng, Koushik Sen, and Ion Stoica. R2e-gym: Procedural environments and hybrid verifiers for scaling open-weights SWE agents. *CoRR*, abs/2504.07164, 2025b. doi: 10.48550/ARXIV.2504.07164. URL https://doi.org/10.48550/arXiv.2504.07164.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by chatgpt really correct? rigorous evaluation of large language models for code generation. *Advances in neural information processing systems*, 36:21558–21572, 2023.

Xialei Liu, Marc Masana, Luis Herranz, Joost van de Weijer, Antonio M Lopez, and Andrew D Bagdanov. Rotate your networks: Better weight consolidation and less catastrophic forgetting. In *Proceedings-International Conference on Pattern Recognition*, 2018.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30, 2017.

Wenhan Ma, Jianyu Wei, Liang Zhao, Hailin Zhang, Bangjun Xiao, Lei Li, Qibin Yang, Bofei Gao, Yudong Wang, Rang Li, Jinhao Dong, Zhifang Sui, and Fuli Luo. MOPD: multi-teacher on-policy distillation for capability integration in LLM post-training. *CoRR*, abs/2606.30406, 2026. doi: 10.48550/ARXIV.2606.30406. URL https://doi.org/10.48550/arXiv.2606.30406.

Arun Mallya, Dillon Davis, and Svetlana Lazebnik. Piggyback: Adapting a single network to multiple tasks by learning to mask weights. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 67–82, 2018.

Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *CoRR*, abs/1803.02999, 2018. URL http://arxiv.org/abs/1803.02999.

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. Training language models to follow instructions with human feedback. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *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*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/b1efde53be364a73914f58805a001731-Abstract-Conference.html.

Valentina Pyatkin, Saumya Malik, Victoria Graf, Hamish Ivison, Shengyi Huang, Pradeep Dasigi, Nathan Lambert, and Hanna Hajishirzi. Generalizing verifiable instruction following. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference. Curran Associates, Inc., 2025. doi: 10.52202/085713-1645. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/46499a0622ecf568b72d17b61e45dbd5-Paper-Datasets_and_Benchmarks_Track.pdf.

Matthew Riemer, Ignacio Cases, Robert Ajemian, Miao Liu, Irina Rish, Yuhai Tu, and Gerald Tesauro. Learning to learn without forgetting by maximizing transfer and minimizing interference. In *International Conference on Learning Representations*, 2019.

Joan Serra, Didac Suris, Marius Miron, and Alexandros Karatzoglou. Overcoming catastrophic forgetting with hard attention to the task. In *International conference on machine learning*, pp. 4548–4557. PMLR, 2018.

Trevor Standley, Amir Zamir, Dawn Chen, Leonidas J. Guibas, Jitendra Malik, and Silvio Savarese. Which tasks should be learned together in multi-task learning? In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 9120–9132. PMLR, 2020. URL http://proceedings.mlr.press/v119/standley20a.html.

Liyuan Wang, Mingtian Zhang, Zhongfan Jia, Qian Li, Chenglong Bao, Kaisheng Ma, Jun Zhu, and Yi Zhong. Afec: Active forgetting of negative transfer in continual learning. *Advances in Neural Information Processing Systems*, 34:22379–22391, 2021.

Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: Theory, method and application. *IEEE transactions on pattern analysis and machine intelligence*, 46(8):5362–5383, 2024.

Mitchell Wortsman, Gabriel Ilharco, Samir Yitzhak Gadre, Rebecca Roelofs, Raphael Gontijo Lopes, Ari S. Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato (eds.), *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pp. 23965–23998. PMLR, 2022. URL https://proceedings.mlr.press/v162/wortsman22a.html.

Bangjun Xiao, Bingquan Xia, Bo Yang, Bofei Gao, Bowen Shen, Chen Zhang, Chenhong He, Chiheng Lou, Fuli Luo, Gang Wang, et al. Mimo-v2-flash technical report. *arXiv preprint arXiv:2601.02780*, 2026.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin A Raffel, and Mohit Bansal. Ties-merging: Resolving interference when merging models. *Advances in neural information processing systems*, 36:7093–7115, 2023.

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

Wenkai Yang, Weijie Liu, Ruobing Xie, Kai Yang, Saiyong Yang, and Yankai Lin. Learning beyond teacher: Generalized on-policy distillation with reward extrapolation. *arXiv preprint arXiv:2602.12125*, 2026.

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pp. 57755–57775. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/yu24p.html.

Yifan Zhang and Team Math-AI. American invitational mathematics examination (aime) 2024, 2024.

Yifan Zhang and Team Math-AI. American invitational mathematics examination (aime) 2025, 2025.

Chujie Zheng, Ziqi Wang, Heng Ji, Minlie Huang, and Nanyun Peng. Model extrapolation expedites alignment. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1025–1041, 2025.

Zhilong Zheng, Letian Tao, Yang Guan, Yujie Yang, Wei Xiong, Kehua Sheng, Bo Zhang, Jingliang Duan, Keqiang Li, and Shengbo Eben Li. Past, future, all at once: Mitigating stability-plasticity dilemma via post-hoc janus rectification, 2026. URL https://arxiv.org/abs/2609.19985.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models, 2023. URL https://arxiv.org/abs/2311.07911.

Luca Zhou, Daniele Solombrino, Donato Crisostomi, Maria Sofia Bucarelli, Giuseppe Alessio D'Inverno, Fabrizio Silvestri, and Emanuele Rodolà. On task vectors and gradients. *CoRR*, abs/2508.16082, 2025. doi: 10.48550/ARXIV.2508.16082. URL https://doi.org/10.48550/arXiv.2508.16082.

A Implementation Details

The implementation details of all baseline methods are summarized as follows:

1. **AvgMerge**: Computes the simple parameter average across all teacher models:

$$\boldsymbol{W}_{1:n}^{\text{AvgMerge}} = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{W}_i$$

2. **AvgMerge + ExPO**: Following Yang et al. (2026), we extrapolate the averaged model parameters by a fraction of λ:

$$\boldsymbol{W}_{1:n}^{\text{ExPO}} = \frac{1+\lambda}{n}\sum_{i=1}^{n}\boldsymbol{W}_i - \lambda\boldsymbol{W}_0$$

   We evaluate $\lambda \in \{0.25, 0.5\}$ and report the performance corresponding to the higher Recovery Rate.

3. **TA**: Direct summation of parameter updates from all teacher models onto the base checkpoint:

$$\boldsymbol{W}_{1:n}^{\text{TA}} = \boldsymbol{W}_0 + \sum_{i=1}^{n}\Delta\boldsymbol{W}_i$$

4. **TA + DARE**: Randomly masks parameter updates with a drop rate p, rescales the remaining updates, and sums them onto the base model:

$$\boldsymbol{W}_{1:n}^{\text{DARE}} = \boldsymbol{W}_0 + \sum_{i=1}^{n}\Delta\widetilde{\boldsymbol{W}}_i$$

$$\Delta\widetilde{\boldsymbol{W}}_i = \frac{1}{1-p}(\mathbf{1}-\boldsymbol{m}_i)\odot\Delta\boldsymbol{W}_i, \quad \boldsymbol{m}_i \sim \text{Bernoulli}(p)$$

5. **Naive MOPD**: On the Open-MOPD benchmark, we report the official baseline results from Gao et al. (2026). On the G-OPD benchmark, we train the student model for 40 steps following the default hyperparameter configuration and original dataset setup from Yang et al. (2026).
6. **ExOPD**: We train the student model for 40 steps using the default hyperparameters and datasets specified in Yang et al. (2026).
7. **Open-MOPD**: We directly evaluate the publicly released checkpoint provided by Gao et al. (2026).[1]

The task orders used to obtain the results in Tab. 1 and 2 are summarized as follows:

1. Task Combination 1: Math $\rightarrow$ Code
2. Task Combination 2: IF $\rightarrow$ Math
3. Task Combination 3: IF $\rightarrow$ Code
4. Task Combination 4: Math $\rightarrow$ IF $\rightarrow$ Code
5. Task Combination 5: Code $\rightarrow$ Math.

B Complete ablation results

Tables 4 and 5 present the complete ablation results on the impact of parameter projection. The results demonstrate that through projection, the model matches the performance of the $\mathcal{T}_1$ teacher while maintaining its performance on $\mathcal{T}_2$ close to that of the $\mathcal{T}_2$ teacher.

[1] https://huggingface.co/BytedTsinghua-SIA/Open-MOPD-SmolLM3-3B-Final

Table 4: Ablation study on the impact of parameter projection on the Open-MOPD benchmark.

Methods	Math (avg@64)			Code (avg@10)			IF			Total
	$AIME_{24}$	$AIME_{25}$	Avg.	LCBv5	LCBv6	Avg.	IFEval	IFB_{test}	Avg.	
Task Combination 1: $\mathcal{T}_1$: Math $\mathcal{T}_2$: Code										
$\mathcal{T}_1$ Expert	23.02	24.69	23.85	-	-	-	-	-	-	-
$\mathcal{T}_2$ Expert	20.63	20.10	20.36	22.63	21.94	22.29	-	-	-	21.32
Distill-Pro	24.27	24.84	24.56	22.16	22.00	22.08	-	-	-	23.32
Distill-Pro-BR	23.80	23.54	23.67	22.10	21.60	21.85	-	-	-	22.76
Distill-Pro-FG	24.64	24.17	24.40	21.98	21.83	21.90	-	-	-	23.15
Distill-Pro-FG-BR	24.48	23.80	24.14	23.11	22.40	22.76	-	-	-	23.45
Task Combination 2: $\mathcal{T}_1$: IF $\mathcal{T}_2$: Math										
$\mathcal{T}_1$ Expert	-	-	-	-	-	-	74.49	24.00	49.25	-
$\mathcal{T}_2$ Expert	23.02	24.69	23.85	-	-	-	65.06	16.67	40.87	32.36
Distill-Pro	24.84	24.27	24.56	-	-	-	74.31	21.33	47.82	36.19
Distill-Pro-BR	23.80	24.95	24.38	-	-	-	75.79	26.00	50.89	37.63
Distill-Pro-FG	24.17	24.79	24.48	-	-	-	75.23	22.67	48.95	36.71
Distill-Pro-FG-BR	23.44	24.32	23.88	-	-	-	73.57	24.00	48.78	36.33
Task Combination 3: $\mathcal{T}_1$: IF $\mathcal{T}_2$: Code										
$\mathcal{T}_1$ Expert	-	-	-	-	-	-	74.49	24.00	49.25	-
$\mathcal{T}_2$ Expert	-	-	-	22.63	21.94	22.29	67.10	19.67	43.38	32.84
Distill-Pro	-	-	-	24.55	23.31	23.93	72.27	22.33	47.30	35.62
Distill-Pro-BR	-	-	-	23.41	23.60	23.51	73.57	23.00	48.28	35.90
Distill-Pro-FG	-	-	-	22.81	22.97	22.89	71.53	23.67	47.60	35.25
Distill-Pro-FG-BR	-	-	-	25.21	22.91	24.06	74.31	27.00	50.65	37.36

Table 5: Ablation study on the impact of parameter projection on the G-OPD benchmark.

Methods	Math (avg@32)					Code (avg@4)				Total
	$AIME_{24}$	$AIME_{25}$	$HMMT_{25}$ (Feb.)	$HMMT_{25}$ (Nov.)	Avg.	HumanEval+	MBPP+	LCBv6	Avg.	
Task Combination 5: $\mathcal{T}_1$: Code $\mathcal{T}_2$: Math										
$\mathcal{T}_1$ Expert	-	-	-	-	-	84.60	69.91	26.14	60.22	-
$\mathcal{T}_2$ Expert	60.42	54.79	31.77	37.81	46.20	81.86	68.72	20.14	56.90	51.55
Distill-Pro	56.35	51.46	31.04	34.79	43.41	82.77	70.83	28.57	60.73	52.07
Distill-Pro-BR	57.29	50.52	29.27	36.67	43.44	82.32	70.57	28.43	60.44	51.94
Distill-Pro-FG	57.50	50.73	30.94	36.25	43.85	82.16	70.04	28.29	60.16	52.01
Distill-Pro-FG-BR	57.40	52.81	30.94	35.73	44.22	82.47	71.43	29.29	61.06	52.64