ADAPTIVE MUTUAL DISTILLATION FOR BALANCED MULTI-TASK POST-TRAINING OF LARGE LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-task post-training of large language models (LLMs) aims to improve performance across tasks with unequal amounts of training data. Existing methods focus primarily on balancing task contributions during single-model training. Different task-balancing strategies can produce models with complementary strengths, creating opportunities for mutual distillation. However, the usefulness of cross-model supervision can vary across tasks, transfer directions, and stages of training. We propose **Adaptive Mutual Distillation (AMD)**, a collaborative post-training framework that jointly trains two models with different task-balancing strategies. AMD evaluates candidate adjustments to distillation weights through short training probes shared across tasks, then uses task-wise validation scores to select an adjustment for each task and transfer direction. Across six benchmarks and three LLM backbones, both AMD models achieve higher average benchmark scores than supervised fine-tuning (SFT) baselines trained with the same sampling strategies. They also outperform the task-balancing methods evaluated in our experiments. Merging the two trained models can further improve their average benchmark score while yielding a single model for inference. The merged models outperform multi-task SFT by an average of 2.91 points across the three backbones.

1 INTRODUCTION

Supervised fine-tuning (SFT) on multi-task instruction mixtures is widely used to adapt large language models (LLMs) to a range of tasks (Sanh et al., 2022; Chung et al., 2022; Wang et al., 2022). These mixtures often contain unequal amounts of data across tasks. Under proportional sampling, tasks with more data are sampled more often. Increasing the emphasis on underrepresented tasks can improve their performance, but may reduce performance on other tasks. Improving overall multi-task performance requires managing these trade-offs.

Existing methods primarily manage these trade-offs by adjusting task contributions within a single model. Data-mixture methods change how training examples are sampled (Xie et al., 2023; Wu et al., 2024; Shin et al., 2026), while optimization-based methods reweight task losses or modify gradients (Chen et al., 2018; Yu et al., 2020; Liu et al., 2023). Multi-task LLM fine-tuning methods also regulate task contributions through convergence feedback, task-specific masks, and task-balanced objectives (Gong et al., 2024; Liu et al., 2024; Zhang et al., 2025). Different strategies place different emphases on tasks and can produce models with complementary strengths. These differences create an opportunity for models to improve through collaboration.

Mutual distillation enables jointly trained models to exchange supervision (Zhang et al., 2018; Anil et al., 2018). The value of this supervision depends on the task and transfer direction. A model may provide useful guidance on one task while benefiting from its partner's guidance on another. These relationships can also change as both models learn. Distillation strength should therefore adapt to the task, transfer direction, and stage of training.

We propose **Adaptive Mutual Distillation (AMD)**, a collaborative post-training framework that jointly trains two models with different task-balancing strategies. In our experiments, one model samples tasks in proportion to their frequencies in the training data. The other uses temperature-smoothed sampling to increase the relative sampling probabilities of tasks with less data. Both models

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.

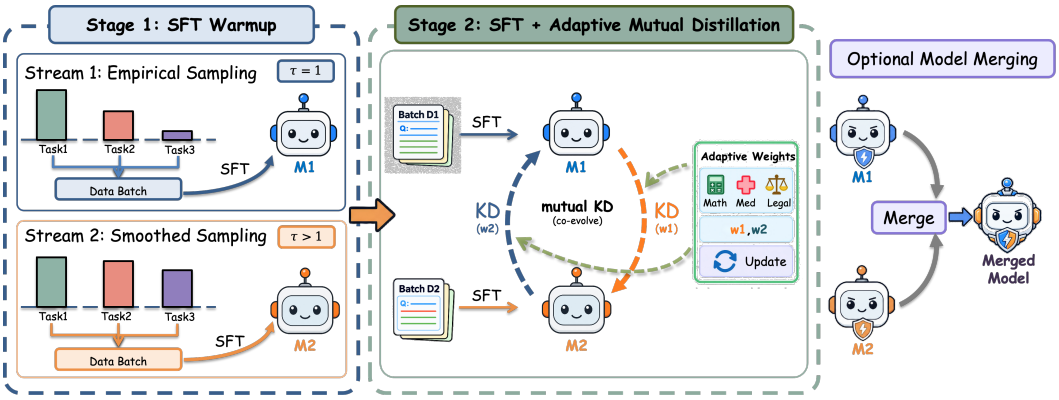


Figure 1: Overview of AMD. Two models with different task-balancing strategies undergo independent SFT warmup, followed by mutual distillation with task- and direction-specific weights adjusted using validation feedback. The trained models can optionally be merged into a single model for inference after collaborative training concludes.

first undergo independent SFT warmup. They then continue training under their respective sampling distributions while exchanging supervision through mutual distillation. Task-wise validation feedback guides the distillation weights for each task and transfer direction. Figure 1 summarizes the two-stage framework and optional post-training model merging.

At each weight update, AMD runs three short training probes from the current model pair, testing whether to decrease, maintain, or increase all distillation weights. It then uses the resulting validation scores to select an adjustment for each task and transfer direction.

We evaluate AMD on three LLM backbones and six benchmarks. On each backbone, both AMD models achieve higher average benchmark scores than SFT baselines trained with the same sampling strategies. They also outperform the task-balancing methods evaluated in our experiments. Merging the two models can further improve performance while yielding a single model for inference. The merged models achieve the highest average benchmark scores among the evaluated methods, exceeding multi-task SFT by 2.70–3.24 points.

Our contributions are as follows:

- We propose AMD, a collaborative framework for multi-task LLM post-training with imbalanced task data. AMD jointly trains models with different task-balancing strategies and enables them to exchange task-wise supervision.
- We develop a validation-guided mechanism that evaluates candidate distillation-weight adjustments through shared training probes and selects updates separately for each task and transfer direction. It adapts supervision throughout training while keeping the number of training probes per update fixed as the number of tasks grows.
- Experiments on three LLM backbones and six benchmarks show that both AMD models outperform their sampling-matched SFT baselines in macro-average performance. Controlled comparisons support adapting distillation weights over training and separately for each task and transfer direction. Model merging provides additional gains while yielding a single model for inference.

2 Related Work

Multi-task learning. Task imbalance and interference are common challenges in multi-task learning. Existing approaches adjust task losses, gradients, or the training data mixture to regulate task contributions. GradNorm balances task training rates through gradient normalization (Chen et al., 2018), PCGrad modifies conflicting gradients (Yu et al., 2020), and FAMO adaptively balances task losses during optimization (Liu et al., 2023). Optimization-based approaches also address task balance in multi-task LLM fine-tuning. CoBa uses convergence signals to adjust task weights (Gong

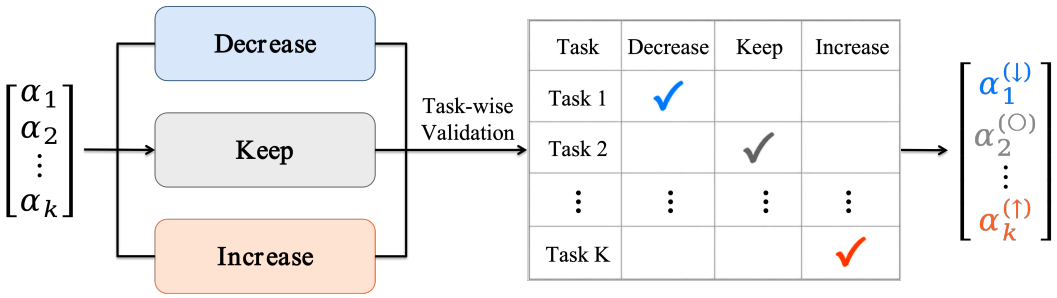


Figure 2: Adaptive distillation-weight update. Three probes globally decrease, retain, or increase the weights. Validation scores determine the weight adjustment for each task and receiving model.

et al., 2024), MFTCoder uses balanced multi-task losses (Liu et al., 2024), and MaZO combines task-specific masks with zeroth-order optimization (Zhang et al., 2025). At the data level, DoReMi (Xie et al., 2023), Mixture-of-Skills (Wu et al., 2024), and DynamixSFT (Shin et al., 2026) control the relative contributions of different data sources. HBO balances data allocation across and within datasets (Wang et al., 2026). AMD brings models with different task-balancing strategies into a collaborative training framework. Each model retains its own strategy while learning from its partner through mutual distillation.

Mutual distillation. Mutual distillation enables concurrently trained models to exchange supervision (Zhang et al., 2018; Anil et al., 2018). OKDDip learns aggregation weights to combine supervision from multiple peers (Chen et al., 2020). Online distillation has also been studied in multi-task learning, where jointly trained single-task networks guide a multi-task network through adaptive feature distillation and task weighting (Jacob et al., 2023). Other approaches adapt how teacher supervision is used. MMKD uses meta-learning to adjust the contributions of multiple pretrained teachers based on feedback from student updates (Zhang et al., 2023). DTO-KD dynamically balances task and distillation losses using their gradients (Hayder et al., 2026). AMD jointly trains two multi-task models with different task-balancing strategies. It uses task-wise validation scores from shared training probes to adjust distillation weights separately for each task and transfer direction as collaborative training progresses.

3 Methodology

AMD uses two training stages: independent SFT warmup and collaborative training with adaptive mutual distillation. Validation feedback guides the distillation weights for each task and transfer direction. Algorithm 1 summarizes the adaptive weight updates used in the second stage. After training, the two models can optionally be merged into a single model for inference (Section 5.4).

3.1 Framework

We consider a training mixture in which each example has a task label. Both models share the same data, architecture, and initialization. We index the tasks participating in mutual distillation by $k \in \{1, \ldots, K\}$. General instruction-following data are included in the training mixture for SFT, but do not participate in mutual distillation.

Stage 1: SFT Warmup. During warmup, each model is trained independently with SFT. One model follows Multi-task SFT (MFT), sampling tasks in proportion to their frequencies in the training mixture. The other follows Smoothed SFT (SSFT), which raises these frequencies to the power $1/\tau_s$, with $\tau_s > 1$, and renormalizes them. This increases the relative sampling probability of less frequent tasks (Arivazhagan et al., 2019; Li et al., 2025b).

Stage 2: SFT with Adaptive Mutual Distillation. After warmup, each model continues training on batches drawn from its own sampling distribution. For tasks participating in mutual distillation, its partner provides token-level predictions on the same examples as distillation targets. Both models are

Algorithm 1 Adaptive update of task-wise directional weights

Input: current model pair and directional weights $\boldsymbol{\alpha}_s$
Output: updated directional weights $\boldsymbol{\alpha}_{s+1}$

for each action $a \in \{\downarrow, \circ, \uparrow\}$ **do**
  Apply a to all $2K$ weights to form $\boldsymbol{\alpha}_s^{(a)}$
  Start an independent probe from the same current model pair
  Run a short training probe using $\boldsymbol{\alpha}_s^{(a)}$
  Record validation scores $R_{k,u}^{(a)}$ for every task k and model u
  Discard parameter changes made during the probe
end for
for each task k and direction $u \leftarrow v$ **do**
  Choose $a_{k,u}^{\star}$ with the largest $R_{k,u}^{(a)}$
  Set $\alpha_{k,s+1}^{u \leftarrow v}$ to its value in $\boldsymbol{\alpha}_s^{(a_{k,u}^{\star})}$
end for
return $\boldsymbol{\alpha}_{s+1}$

updated through their own training objectives, so the supervision they provide evolves during training and reflects their current token-level predictions.

Let $u \leftarrow v$ denote distillation from model v to model u. For task k, the receiving model uses the following training objective:

$$\mathcal{L} = \big(1 - \alpha_k^{u \leftarrow v}\big)\mathcal{L}_{\mathrm{SFT}} + \alpha_k^{u \leftarrow v}\mathcal{L}_{\mathrm{KD}}.$$

Here, $\mathcal{L}_{\mathrm{SFT}}$ is the supervised fine-tuning loss. $\mathcal{L}_{\mathrm{KD}}$ is the KL divergence from the supervising model's token distribution to the receiving model's, averaged over response tokens (Hinton et al., 2015). The weight $\alpha_k^{u \leftarrow v} \in [0, 1]$ controls the distillation strength for this task and direction.

Fixed-weight distillation. Under this objective, we consider two fixed-weight variants for comparison with AMD. During Stage 2, FIXED-BI uses the same fixed distillation weight α_{Bi} for all participating tasks and both transfer directions, so each model receives supervision from its partner. At each evaluation point in Stage 2, FIXED-UNI selects the model with the higher validation score on each participating task as the teacher for that task. The teacher supervises its partner with the fixed weight α_{Uni} until the next evaluation.

3.2 ADAPTIVE-WEIGHT DISTILLATION

AMD adjusts the distillation weight for each task and transfer direction as training progresses. Shared training probes evaluate candidate adjustments, and the receiving model's task-specific validation scores guide the selection of each update. Figure 2 illustrates this process for one transfer direction.

Global probing. At update round s, the vector $\boldsymbol{\alpha}_s$ contains the $2K$ task- and direction-specific distillation weights. AMD constructs three candidate configurations by decreasing, keeping, or increasing all entries of this vector. These actions are denoted by $a \in \{\downarrow, \circ, \uparrow\}$. Distillation weights are bounded mixing coefficients. We therefore apply increases and decreases in logit space and map the updated values back through the sigmoid function. This keeps the weights within the valid range and yields smaller changes near the boundaries. The logit-space step size is δ_s; the keep action leaves the weights unchanged.

Each candidate is evaluated through a short training probe starting from the same current model pair. During each probe, both models train on shared batches using the candidate weights (Appendix A.2). Evaluation yields a validation score $R_{k,u}^{(a)}$ for each task k and model u. Parameter changes made during the probe are then discarded.

Task-wise directional selection. For each task k and direction $u \leftarrow v$, AMD selects the action with the highest validation score $R_{k,u}^{(a)}$ for the receiving model u. The corresponding candidate weight

Table 1: Main results on Qwen3-0.6B-Base. Avg. is the unweighted mean across six benchmarks. Δ Avg. is measured against Smoothed SFT for $\tau > 1$ rows and Multi-task SFT otherwise, including + Merged. Bold indicates column-wise best results.

Method	Sampling	Math		Medicine		Law		Avg.	Δ Avg.
		GSM8K	Math-500	MedQA	MedBullets	LawBench	CMMLU-Law		
SFT Baselines									
Multi-task SFT	$\tau = 1$	57.47	42.40	27.34	40.10	29.53	48.99	40.97	–
Smoothed SFT	$\tau > 1$	57.47	34.20	30.71	44.16	28.48	50.03	40.84	–
Prior Methods									
PCGrad	$\tau = 1$	56.10	34.40	29.38	47.40	32.86	48.64	41.46	+0.49
FAMO	$\tau = 1$	57.39	37.40	30.40	43.83	34.76	48.41	42.03	+1.06
CoBa	$\tau = 1$	56.56	41.40	29.46	46.43	26.48	49.41	41.62	+0.65
MFTCoder	$\tau = 1$	58.38	43.40	29.38	45.13	27.17	48.88	42.06	+1.09
HBO	$\tau = 1$	60.42	33.60	29.54	42.05	**34.93**	49.58	41.69	+0.72
Our Methods									
Fixed-Bi	$\tau = 1$	58.30	41.60	25.61	45.78	28.22	50.64	41.69	+0.72
	$\tau > 1$	60.35	41.00	28.75	44.32	29.56	50.79	42.46	+1.62
Fixed-Uni	$\tau = 1$	60.27	43.20	24.98	43.18	26.88	50.20	41.45	+0.48
	$\tau > 1$	57.39	41.40	**33.94**	48.70	27.96	49.76	43.19	+2.35
Adaptive	$\tau = 1$	60.05	**44.00**	30.40	44.97	31.40	49.85	43.45	+2.48
	$\tau > 1$	60.65	40.80	28.91	46.43	31.66	50.67	43.19	+2.35
+ Merged		**61.11**	42.60	28.67	**49.03**	29.90	**51.17**	**43.75**	**+2.78**

becomes $\alpha^{u \leftarrow v}_{k,s+1}$. The resulting vector $\boldsymbol{\alpha}_{s+1}$ can therefore combine weights from different candidates. The original model pair resumes training with these updated weights. Algorithm 1 summarizes the procedure used at each adaptive update round.

4 Experiments

We evaluate each AMD branch against its sampling-matched SFT baseline, fixed-strength distillation variants, and prior task-balancing methods. Tables 1–3 report these comparisons across the evaluated backbones, together with the results of post-training model merging.

4.1 Setup

Experimental setting. We evaluated AMD on Qwen3-0.6B-Base (Yang et al., 2025), Llama-3.1-8B (Grattafiori et al., 2024), and Llama-3.2-1B (Meta AI, 2024). The training mixture covered mathematics, medicine, law, and general instruction following. It contained 100k examples from NuminaMath (AI-MO, 2025), 9,178 from MedQA (Jin et al., 2021), 50k from DISC-Law (Yue et al., 2023), and 10k from InfinityInstruct (Li et al., 2025a).

The two AMD models shared this mixture. One used empirical sampling ($\tau = 1$), and the other used temperature-smoothed sampling ($\tau_s = 3$). We used full-parameter SFT with a learning rate of 1×10^{-5} for Qwen3-0.6B-Base and Llama-3.2-1B. For Llama-3.1-8B, we used LoRA with rank 16, alpha 32, dropout 0.05, and the target modules set to `all`. The learning rate was 5×10^{-5}. In the main experiments, all models were trained for three epochs with an effective batch size of 64. Mutual distillation began at step 3000 with temperature $T = 1.0$.

Distillation configurations. Across all three backbones, Fixed-Bi and Fixed-Uni used fixed distillation weights of 0.4 and 0.2, respectively. Adaptive initialized all task- and direction-specific weights to 0.4 and updated them within $[0, 1]$. Adaptive used a logit-space step size of $\delta_s = \mu_s \delta_0$, with $\delta_0 = 0.5$. We set $\mu_s = \sqrt{1 - t/T_{\max}}$, where $t/T_{\max}$ is the fraction of training completed. After warmup, we updated the distillation weights every 1000 training steps. Fixed-Uni used the same interval for teacher selection. Each Adaptive probe used 1% of the one-epoch training budget. Adaptive updates used task-wise validation accuracy.

Table 2: Main results on Llama-3.1-8B.

Method	Sampling	Math		Medicine		Law		Avg.	Δ Avg.
		GSM8K	Math-500	MedQA	MedBullets	LawBench	CMMLU-Law		
SFT Baselines									
Multi-task SFT	$\tau = 1$	58.45	27.20	49.25	51.14	28.46	**50.49**	44.16	–
Smoothed SFT	$\tau > 1$	50.42	26.00	50.67	55.19	35.84	47.99	44.35	–
Prior Methods									
PCGrad	$\tau = 1$	62.32	25.80	50.12	55.36	31.93	49.92	45.91	+1.75
FAMO	$\tau = 1$	62.17	22.80	50.59	54.71	31.25	47.53	44.84	+0.68
CoBa	$\tau = 1$	61.87	29.00	51.69	51.46	20.35	48.51	43.81	-0.35
MFTCoder	$\tau = 1$	63.61	26.20	**52.08**	52.11	28.40	48.96	45.23	+1.07
HBO	$\tau = 1$	**66.57**	21.00	**52.08**	**56.66**	33.08	45.01	45.73	+1.57
Our Methods									
Fixed-Bi	$\tau = 1$	61.49	26.80	50.59	53.73	28.82	50.22	45.27	+1.11
	$\tau > 1$	62.93	27.80	48.86	56.01	33.46	47.60	46.11	+1.76
Fixed-Uni	$\tau = 1$	63.00	24.20	51.37	53.90	33.32	47.76	45.59	+1.43
	$\tau > 1$	64.06	29.60	49.96	55.03	32.89	47.10	46.44	+2.09
Adaptive	$\tau = 1$	63.61	27.60	51.69	52.60	34.17	50.30	46.66	+2.50
	$\tau > 1$	63.23	29.40	49.65	54.87	**36.41**	47.79	46.89	+2.54
+ Merged		65.13	**29.80**	51.37	55.52	33.68	48.92	**47.40**	**+3.24**

Table 3: Main results on Llama-3.2-1B.

Method	Sampling	Math		Medicine		Law		Avg.	Δ Avg.
		GSM8K	Math-500	MedQA	MedBullets	LawBench	CMMLU-Law		
SFT Baselines									
Multi-task SFT	$\tau = 1$	19.94	7.00	30.40	29.55	16.55	33.53	22.83	–
Smoothed SFT	$\tau > 1$	20.39	6.80	28.20	32.79	16.70	32.94	22.97	–
Prior Methods									
PCGrad	$\tau = 1$	15.62	5.80	27.65	35.88	22.15	32.20	23.22	+0.39
FAMO	$\tau = 1$	21.99	6.80	30.95	29.55	21.53	30.43	23.54	+0.71
CoBa	$\tau = 1$	18.42	8.20	27.42	33.12	21.03	31.93	23.35	+0.52
MFTCoder	$\tau = 1$	19.33	9.00	26.00	34.90	20.86	30.81	23.48	+0.65
HBO	$\tau = 1$	17.51	6.20	29.22	**36.36**	20.25	34.04	23.93	+1.10
Our Methods									
Fixed-Bi	$\tau = 1$	23.58	7.60	31.50	33.44	16.41	31.90	24.07	+1.24
	$\tau > 1$	23.65	8.80	**32.05**	30.19	17.25	31.57	23.92	+0.95
Fixed-Uni	$\tau = 1$	21.91	7.60	29.38	32.31	21.29	33.79	24.38	+1.55
	$\tau > 1$	20.55	**9.20**	29.77	33.60	**23.82**	30.11	24.51	+1.54
Adaptive	$\tau = 1$	**25.32**	7.60	28.83	34.58	20.93	32.92	25.03	+2.20
	$\tau > 1$	22.44	7.80	26.87	34.09	21.76	**34.29**	24.54	+1.57
+ Merged		24.56	7.20	31.89	35.88	20.69	32.98	**25.53**	**+2.70**

Evaluation. The evaluation covered six benchmarks across three domains: GSM8K (Cobbe et al., 2021) and Math-500 (Hendrycks et al., 2021; Lightman et al., 2024) for mathematics, MedQA (Jin et al., 2021) and MedBullets (Chen et al., 2025) for medicine, and LawBench (Fei et al., 2024) and CMMLU-Law (Li et al., 2024) for law. All evaluations used OpenCompass v0.5.3 (OpenCompass Contributors, 2023). Overall multi-task performance was measured by the macro-average score, defined as the unweighted mean of the six benchmark scores. Checkpoints were selected using validation performance; the validation and model-selection protocols are detailed in Appendix A.1.

Comparisons. We compared AMD with MFT and SSFT baselines trained independently using the same sampling distributions as the corresponding AMD models. We also evaluated prior single-model task-balancing methods: PCGrad (Yu et al., 2020), FAMO (Liu et al., 2023), CoBa (Gong et al., 2024),

Table 4: Component comparisons on Llama-3.2-1B. Scores are six-benchmark macro-averages. Sampling lists the two temperatures; paired scores follow the same model order. Task-Tied MD selects one action per direction, shared across tasks.

Method	Sampling	Avg.
SFT baselines	$(1, 3)$	22.83 / 22.97
Same-temp MD	$(1, 1)$	22.50 / 22.45
Same-temp MD	$(3, 3)$	23.18 / 22.22
Fixed-Bi	$(1, 3)$	24.07 / 23.92
Adaptive	$(1, 3)$	**25.03 / 24.54**

(a) Sampling configurations

Method	Sampling	Avg.
Globally Tied MD	$(3, 3)$	23.05 / 23.41
Final-Weight MD	$(1, 3)$	23.41 / 22.01
Direction-Tied MD	$(1, 3)$	24.59 / 23.38
Task-Tied MD	$(1, 3)$	24.33 / 23.69
Adaptive	$(1, 3)$	**25.03 / 24.54**

(b) Distillation-weight policies

MFTCoder (Liu et al., 2024), and HBO (Wang et al., 2026). These methods began from the empirical task mixture. Fixed-Bi and Fixed-Uni provided fixed-strength comparisons for adaptive distillation. Within each backbone, all methods used the same training examples, tokenizer, generation settings, and evaluation protocol.

4.2 Main Results

Tables 1–3 compare AMD with SFT baselines and prior task-balancing methods across three LLM backbones. On all three backbones, both Adaptive branches achieve higher macro-average scores than their corresponding SFT baselines. They also outperform all evaluated prior task-balancing methods. Relative to the corresponding SFT baselines, the gains range from 2.20 to 2.50 points for the empirical-sampling branch and from 1.57 to 2.54 points for the smoothed-sampling branch.

Among the distillation variants, Adaptive produces the individual model with the highest macro-average score on each backbone. Its highest-scoring branch exceeds the best fixed-distillation branch by 0.26, 0.45, and 0.52 points on Qwen3-0.6B-Base, Llama-3.1-8B, and Llama-3.2-1B, respectively. The gains vary across sampling branches: on Qwen3-0.6B-Base, Adaptive and Fixed-Uni tie under smoothed sampling.

Merging the two Adaptive models further improves macro-average performance. The merged models achieve the highest macro-average scores in all three main tables, reaching 43.75, 47.40, and 25.53 on Qwen3-0.6B-Base, Llama-3.1-8B, and Llama-3.2-1B, respectively. These scores exceed those of the better Adaptive branch by 0.30, 0.51, and 0.50 points, respectively. Across the three backbones, the merged models improve over multi-task SFT by an average of 2.91 points.

5 Analysis

We analyze AMD on Llama-3.2-1B, starting with sampling configurations and adaptive distillation. We then examine variability across seeds and the additional gains from model merging. Appendices C–E examine fixed distillation weights, weight-update schedules, and training budgets.

5.1 Effect of Sampling Distributions

Table 4(a) compares mutual distillation when the two models use the same or different sampling distributions. Same-temp MD uses the same fixed bidirectional distillation weights as Fixed-Bi, with different sampling seeds for the two models. Under empirical sampling, the two models achieve 22.50 and 22.45, both below the corresponding SFT baseline of 22.83. Under smoothed sampling, they achieve 23.18 and 22.22, with one above and the other below the corresponding SFT baseline of 22.97. With different sampling distributions, Fixed-Bi reaches 24.07 and 23.92, outperforming both corresponding SFT baselines. Its highest branch score exceeds the best same-temperature result by 0.89 points. These results support using different sampling distributions for the two models under fixed bidirectional distillation.

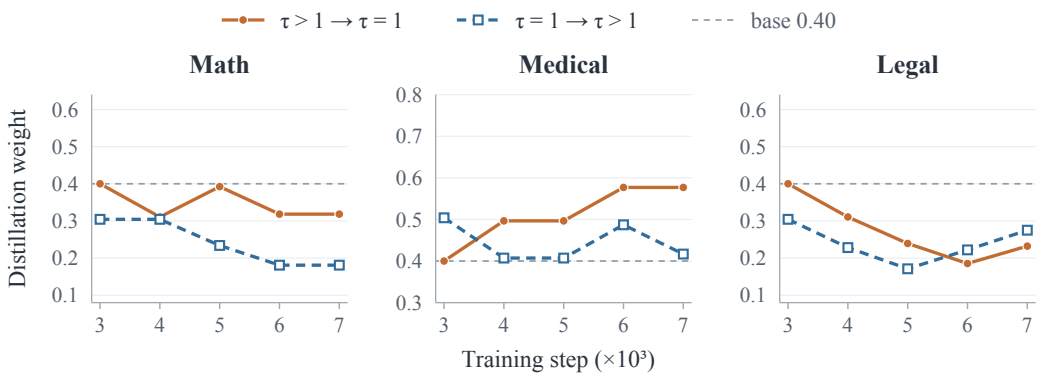


Figure 3: Task- and direction-specific distillation-weight trajectories on Llama-3.2-1B. The first point in each trajectory shows the weight after the initial adaptive update at step 3000.

5.2 Effect of Adaptive Weight Updates

Online adaptation yields higher scores in both branches than keeping the learned final weights fixed throughout collaborative training. Following the same SFT warmup, Final-Weight MD keeps Adaptive's final task- and direction-specific distillation weights fixed throughout Stage 2. It achieves 23.41 and 22.01 under empirical and smoothed sampling, respectively, compared with 25.03 and 24.54 for Adaptive (Table 4(b)).

Updating distillation weights separately for each transfer direction yields higher scores in both branches than Direction-Tied MD. Direction-Tied MD retains task-specific adaptation but ties the two directional weights within each task. At each update, it averages the two branches' task-specific validation scores for each candidate action to select a shared action for both directions. It achieves 24.59 under empirical sampling and 23.38 under smoothed sampling (Table 4(b)).

Updating distillation weights separately for each task yields higher scores in both branches than Task-Tied MD. Task-Tied MD retains direction-specific adaptation but selects a single update action per direction and applies it to all tasks. It achieves 24.33 under empirical sampling and 23.69 under smoothed sampling (Table 4(b)).

To assess whether a shared update action benefits both models under matched smoothed sampling, we compare Globally Tied MD with fixed-weight Same-temp MD. Globally Tied MD selects one action across tasks and transfer directions using a weighted average of task-wise validation scores. Its best branch reaches 23.41, compared with the best Same-temp MD branch at 23.18 (Table 4(a,b)). The shared update improves one branch but slightly lowers the other relative to Same-temp MD.

To inspect how task- and direction-specific weights evolve during training, Figure 3 shows trajectories for Math, Medicine, and Law. In Math, both weights decrease overall, with a generally higher weight for the smoothed-to-empirical direction. In Medicine, this direction increases while the reverse direction fluctuates, changing their relative ordering. In Law, both weights decrease initially and then rise toward the end of training. These different trajectories illustrate how Adaptive changes distillation strength by task and transfer direction.

5.3 Variability Across Training Seeds

To assess whether the gains from adaptive distillation persist across training seeds, we repeat each distillation regime with two additional seeds, yielding three runs per regime (Table 5).

Adaptive achieves 24.60 ± 0.38 under empirical sampling and 24.59 ± 0.10 under smoothed sampling. Both branches outperform their corresponding Fixed-Bi and Fixed-Uni baselines on average. The smoothed-sampling branch also exceeds both corresponding baselines in every reported run. The averaged gains for both branches and the per-run gains under smoothed sampling support the repeatability of AMD's performance improvements across training seeds.

Table 5: Training-seed variability of the three AMD regimes on Llama-3.2-1B. Scores are unweighted six-benchmark macro-averages, summarized across three runs as mean ± sample standard deviation. The + Merged row reports the merged Adaptive model from each run.

Method	Sampling	Macro Avg. by seed			Mean ± Std.
		42	43	44	
FIXED-BI	$\tau = 1$	24.07	23.79	23.83	23.90 ± 0.15
	$\tau = 3$	23.92	24.14	24.44	24.17 ± 0.26
FIXED-UNI	$\tau = 1$	24.38	24.52	24.16	24.35 ± 0.18
	$\tau = 3$	24.51	24.40	24.45	24.45 ± 0.06
Adaptive	$\tau = 1$	25.03	24.32	24.44	24.60 ± 0.38
	$\tau = 3$	24.54	24.70	24.52	24.59 ± 0.10
+ Merged		25.53	25.61	25.28	25.47 ± 0.17

5.4 POST-TRAINING MODEL MERGING

We evaluate post-training model merging as a way to obtain additional gains from the two AMD models while using a single model for inference. Using MergeKit (Goddard et al., 2024), we average the parameters of the two models from the same training step with equal weights. We apply the same procedure to the independently trained Multi-task SFT and Smoothed SFT models.

Table 6: Model merging on Llama-3.2-1B. Scores are six-benchmark macro-averages; Δ Merge is the gain over the better model before merging.

Training	Empirical sampling	Smoothed sampling	Merged	Δ Merge
SFT baselines	22.83	22.97	24.08	+1.11
AMD	25.03	24.54	**25.53**	+0.50

The merged SFT baseline provides a reference for assessing the benefit of collaborative training beyond averaging independently trained models. Both AMD branches outperform this baseline, which scores 24.08 (Table 6). Their advantage is therefore already present before merging, supporting the contribution of collaborative training to the performance of the constituent models.

Merging further raises AMD's score from 25.03 for its better branch to 25.53, exceeding the merged SFT baseline by 1.45 points. Across three training seeds, the merged AMD model achieves 25.47 ± 0.17 and outperforms both constituent models in every run (Table 5). Parameter averaging thus builds on the gains from collaborative training and combines the two models into a stronger model for inference with the same backbone architecture.

6 CONCLUSION

AMD addresses imbalanced multi-task LLM post-training by enabling models with different sampling distributions to learn from each other. Validation feedback guides the distillation strength for each task and transfer direction throughout training. Across three LLM backbones and six benchmarks, both trained models outperform their sampling-matched SFT baselines in macro-average performance. Optional parameter merging provides further gains and produces a single model for inference. The merged models achieve the highest macro-average scores among the compared methods, exceeding multi-task SFT by 2.70–3.24 points. These results support adaptive collaboration between models with different task emphases for improving overall multi-task performance.

AI USE STATEMENT

Generative AI tools were used only to polish the language of the manuscript. They were not used to assist with translation or to interpret results. The authors take responsibility for the final content.

REPRODUCIBILITY STATEMENT

The training framework and adaptive weight updates are described in Section 3 and Algorithm 1. Section 4.1 and Appendix A document the datasets, backbone models, training settings, evaluation protocol, and checkpoint-selection procedure. Section 5.3 reports variability across training seeds.

REFERENCES

AI-MO. NuminaMath-1.5. https://huggingface.co/datasets/AI-MO/NuminaMath-1.5, 2025. Hugging Face dataset.

Rohan Anil, Gabriel Pereyra, Alexandre Passos, Robert Ormandi, George E. Dahl, and Geoffrey E. Hinton. Large scale distributed neural network training through online distillation. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rkr1UDeC-.

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George Foster, Colin Cherry, Wolfgang Macherey, Zhifeng Chen, and Yonghui Wu. Massively multilingual neural machine translation in the wild: Findings and challenges, 2019. URL https://arxiv.org/abs/1907.05019.

Defang Chen, Jian-Ping Mei, Can Wang, Yan Feng, and Chun Chen. Online knowledge distillation with diverse peers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 3430–3437, 2020. doi: 10.1609/aaai.v34i04.5746. URL https://ojs.aaai.org/index.php/AAAI/article/view/5746.

Hanjie Chen, Zhouxiang Fang, Yash Singla, and Mark Dredze. Benchmarking large language models on answering and explaining challenging medical questions. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 3563–3599, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.naacl-long.182. URL https://aclanthology.org/2025.naacl-long.182/.

Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. GradNorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 794–803. PMLR, 2018. URL https://proceedings.mlr.press/v80/chen18a.html.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models, 2022. URL https://arxiv.org/abs/2210.11416.

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

Zhiwei Fei, Xiaoyu Shen, Dawei Zhu, Fengzhe Zhou, Zhuo Han, Alan Huang, Songyang Zhang, Kai Chen, Zhixin Yin, Zongwen Shen, Jidong Ge, and Vincent Ng. LawBench: Benchmarking legal knowledge of large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 7933–7962, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.452. URL https://aclanthology.org/2024.emnlp-main.452/.

Charles Goddard, Shamane Siriwardhana, Malikeh Ehghaghi, Luke Meyers, Vladimir Karpukhin, Brian Benedict, Mark McQuade, and Jacob Solawetz. Arcee's MergeKit: A toolkit for merging large language models. In Franck Dernoncourt, Daniel Preoţiuc-Pietro, and Anastasia Shimorina (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pp. 477–485, Miami, Florida, US, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-industry.36. URL https://aclanthology.org/2024.emnlp-industry.36/.

Zi Gong, Hang Yu, Cong Liao, Bingchang Liu, Chaoyu Chen, and Jianguo Li. CoBa: Convergence balancer for multitask finetuning of large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 8063–8077, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.459. URL https://aclanthology.org/2024.emnlp-main.459/.

Aaron Grattafiori et al. The Llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Zeeshan Hayder, Ali Cheraghian, Lars Petersson, Mehrtash Harandi, and Richard Hartley. DTO-KD: Dynamic trade-off optimization for effective knowledge distillation. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/e546ce4230c15bc422a500bb47b341ea-Abstract-Conference.html.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper_files/paper/2021/hash/be83ab3ecd0db773eb2dc1b0a17836a1-Abstract-round2.html.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015. URL https://arxiv.org/abs/1503.02531.

Geethu Miriam Jacob, Vishal Agarwal, and Björn Stenger. Online knowledge distillation for multi-task learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 2359–2368, 2023. URL https://openaccess.thecvf.com/content/WACV2023/html/Jacob_Online_Knowledge_Distillation_for_Multi-Task_Learning_WACV_2023_paper.html.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14):6421, 2021. doi: 10.3390/app11146421. URL https://doi.org/10.3390/app11146421.

Haonan Li, Yixuan Zhang, Fajri Koto, Yifei Yang, Hai Zhao, Yeyun Gong, Nan Duan, and Timothy Baldwin. CMMLU: Measuring massive multitask language understanding in Chinese. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 11260–11285, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.671. URL https://aclanthology.org/2024.findings-acl.671/.

Jijie Li, Li Du, Hanyu Zhao, Bo-wen Zhang, Liangdong Wang, Boyan Gao, Guang Liu, and Yonghua Lin. Infinity instruct: Scaling instruction selection and synthesis to enhance language models, 2025a. URL https://arxiv.org/abs/2506.11116.

Tianjian Li, Haoran Xu, Weiting Tan, Kenton Murray, and Daniel Khashabi. Upsample or upweight? balanced training on heavily imbalanced datasets. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 3325–3343, Albuquerque, New Mexico, April 2025b. Association for Computational Linguistics. doi: 10.18653/v1/2025.naacl-long.171. URL https://aclanthology.org/2025.naacl-long.171/.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=v8L0pN6EOi.

Bingchang Liu, Chaoyu Chen, Zi Gong, Cong Liao, Huan Wang, Zhichao Lei, Ming Liang, Dajun Chen, Min Shen, Hailian Zhou, Wei Jiang, Hang Yu, and Jianguo Li. MFTCoder: Boosting code LLMs with multitask fine-tuning. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 5430–5441, 2024. doi: 10.1145/3637528.3671609. URL https://doi.org/10.1145/3637528.3671609.

Bo Liu, Yihao Feng, Peter Stone, and Qiang Liu. FAMO: Fast adaptive multitask optimization. In *Advances in Neural Information Processing Systems*, pp. 57226–57243, 2023. URL https://openreview.net/forum?id=zMeemcUeXL.

Meta AI. Llama 3.2 1B model card. https://huggingface.co/meta-llama/Llama-3.2-1B, 2024. Hugging Face model card.

OpenCompass Contributors. OpenCompass: A universal evaluation platform for foundation models. https://github.com/open-compass/opencompass, 2023.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, et al. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=9Vrb9D0WI4.

Haebin Shin, Lei Ji, Xiao Liu, Zhiwei Yu, Hyunwoo Yoo, Qi Chen, and Yeyun Gong. DynamixSFT: Dynamic mixture optimization of instruction tuning collections. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 39590–39603. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.findings-acl.1972. URL https://aclanthology.org/2026.findings-acl.1972/.

Weixuan Wang, Minghao Wu, Barry Haddow, and Alexandra Birch. HBO: Hierarchical balancing optimization for fine-tuning large language models. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=JnhahbMvRE.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, et al. Super-NaturalInstructions: Generalization via declarative instructions on 1600+ NLP tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 5085–5109, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.340. URL https://aclanthology.org/2022.emnlp-main.340/.

Minghao Wu, Thuy-Trang Vu, Lizhen Qu, and Gholamreza Haffari. Mixture-of-skills: Learning to optimize data usage for fine-tuning large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 14226–14240, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.787. URL https://aclanthology.org/2024.emnlp-main.787/.

Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy Liang, Quoc V. Le, Tengyu Ma, and Adams Wei Yu. DoReMi: Optimizing data mixtures speeds up language model pretraining, 2023. URL https://arxiv.org/abs/2305.10429.

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, 2025. URL https://arxiv.org/abs/2505.09388.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In *Advances in Neural Information Processing Systems*, pp. 5824–5836, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/3fe78a8acf5fda99de95303940a2420c-Abstract.html.

Shengbin Yue, Wei Chen, Siyuan Wang, Bingxuan Li, Chenchen Shen, Shujun Liu, Yuxuan Zhou, Yao Xiao, Song Yun, Xuanjing Huang, and Zhongyu Wei. DISC-LawLLM: Fine-tuning large language models for intelligent legal services, 2023. URL https://arxiv.org/abs/2309.11325.

Hailin Zhang, Defang Chen, and Can Wang. Adaptive multi-teacher knowledge distillation with meta-learning. In *2023 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1943–1948, 2023. doi: 10.1109/ICME55011.2023.00333. URL https://arxiv.org/abs/2306.06634.

Ying Zhang, Tao Xiang, Timothy M. Hospedales, and Huchuan Lu. Deep mutual learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4320–4328, June 2018. URL https://openaccess.thecvf.com/content_cvpr_2018/html/Zhang_Deep_Mutual_Learning_CVPR_2018_paper.html.

Zhen Zhang, Yifan Yang, Kai Zhen, Nathan Susanj, Athanasios Mouchtaris, Siegfried Kunzmann, and Zheng Zhang. MaZO: Masked zeroth-order optimization for multi-task fine-tuning of large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 18526–18543, Suzhou, China, November 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.emnlp-main.935. URL https://aclanthology.org/2025.emnlp-main.935/.

APPENDIX

The appendix provides experimental details and additional analyses of temperature-smoothed baselines, fixed distillation weights, weight-update schedules, and training budgets.

A EXPERIMENTAL PROTOCOL

A.1 DATA MIXTURE AND EVALUATION

The training mixture assigns one fixed task category to each example before training. Table 7 summarizes the category-level datasets and sample caps used in the main experiments. These task categories are used consistently by the sampler, the single-model multi-task baselines, and the AMD distillation policies.

Table 7: Training mixture for the main experiments. Each row lists one task category, its source dataset, and the configured sample cap.

Category	Dataset	Cap
Math	NuminaMath-1.5 (AI-MO, 2025)	100k
Medicine	MedQA (Jin et al., 2021)	9,178
Law	DISC-Law (Yue et al., 2023)	50k
General	InfinityInstruct (Li et al., 2025a)	10k

For evaluation, all results reported within a backbone family use the same six benchmarks and post-processing rules. We use two benchmarks for each of the mathematical reasoning, medical reasoning, and legal reasoning categories. The reported aggregate score is the macro-average over these six benchmark scores. LawBench evaluation covered the full benchmark, and CMMLU-Law included all law-related subsets of CMMLU. Training, validation, and test data were mutually disjoint.

Validation and model selection. Adaptive updates used accuracy on validation subsets from the validation sets provided with MATH, MedQA, and LawBench for the mathematics, medicine, and law tasks, respectively. We randomly sampled 100 validation examples for mathematics, 100 for medicine, and 200 for law. These subsets remained fixed throughout training and were shared across all three backbones. For each task and receiving model, we use the accuracy returned by OpenCompass on the corresponding fixed validation subset as $R_{k,u}^{(a)}$ to compare the candidate probes. General instruction-following data were used for SFT only and did not contribute to adaptive validation feedback. For AMD, the checkpoint was selected by taking the unweighted mean of the six validation accuracies across both branches and the three tasks. The training step with the highest mean was selected, and both models at that step were used for branch-wise reporting. The same pair was then merged to produce the + Merged result, without a separate search over merged candidates or further training. Single-model methods used the checkpoint with the highest mean validation score.

A.2 TRAINING CONFIGURATION

Unless a method requires a different batch structure, we keep the data mixture, tokenizer, cutoff length, learning-rate schedule, task-utility mapping, and six-benchmark evaluation protocol fixed within each backbone family. Training runs are conducted on 8×A800-SXM4-80GB GPUs. The main Llama-3.2-1B and Qwen3-0.6B-Base runs use full-parameter SFT with sequence cutoff length 2048, bf16 training, cosine scheduling, warmup ratio 0.1, and learning rate $1e{-5}$. The Llama-3.1-8B runs use LoRA rather than full-parameter updates due to the larger backbone size.

The two-model AMD setup uses two task-sampling temperatures: $\tau = 1$ preserves the empirical task mixture, while $\tau > 1$ increases the relative sampling probability of smaller task categories. In the reported main runs, the two models use $\tau = 1$ and $\tau = 3$. AMD enables cross-model distillation after the initial SFT phase; the main policy runs start distillation at step 3000 and use distillation temperature $T = 1.0$. Method-specific policy settings are fixed before test evaluation and are read from the corresponding training configuration rather than tuned per reported benchmark.

Each probe uses the next batches that would be sampled along the current training trajectory. Both models share these batches, and all candidates use the same batch sequence. Model, optimizer, learning-rate scheduler, data iterator, and random number generator states are restored to their pre-probe values before each candidate and before training resumes with the selected weights. When validation scores tie, Adaptive leaves the corresponding distillation weight unchanged, while Fixed-Uni retains the current teacher direction.

To keep the effective optimizer-update scale comparable, standard two-model runs use two models with per-device batch size 4 and gradient accumulation 2. MFTCoder and CoBa use per-device batch size 8 and gradient accumulation 1, while PCGrad and FAMO use task-grouped micro-batches so that each optimizer update covers all four task categories. This keeps the comparison data-matched, although not exactly wall-clock-cost matched.

A.3 Rationale for Shared Probing and Task-wise Selection

AMD uses shared training probes to obtain task-specific feedback on candidate distillation-weight adjustments. Each probe starts from the same current model and optimizer states and applies an adjustment to the full weight vector. Its validation results measure each receiving model's performance on each task after training under that candidate configuration. Selecting a single action from an aggregate score forces all tasks and transfer directions to share the same adjustment, even when their validation scores favor different candidates. AMD preserves these differences by using each task's validation ranking to select the corresponding directional weight adjustment.

The selected adjustments are combined into a new weight vector, and training resumes from the original state with these weights. This feedback-guided recombination enables task- and direction-specific updates while keeping the number of training probes per update fixed as the number of tasks grows. Because the combined vector can differ from every probed configuration, its performance need not match the individual probe scores. The probe results provide guidance for constructing the update, while the resulting training performance determines its effectiveness.

For each transfer direction, Task-Tied MD computes the unweighted mean validation score across tasks for each candidate action. It selects the action with the highest mean score and applies it to all tasks in that direction.

B Temperature-Smoothed Baseline Setting

Table 8 reports a sampling-matched $\tau > 1$ control on Llama-3.2-1B. The table includes the temperature-smoothed SFT reference, prior single-model baselines, and the corresponding smoothed-sampling branches of the mutual-distillation regimes. Each reported model is trained with the same temperature-smoothed sampling distribution. The table uses the same six benchmark columns and macro-average definition as Table 3.

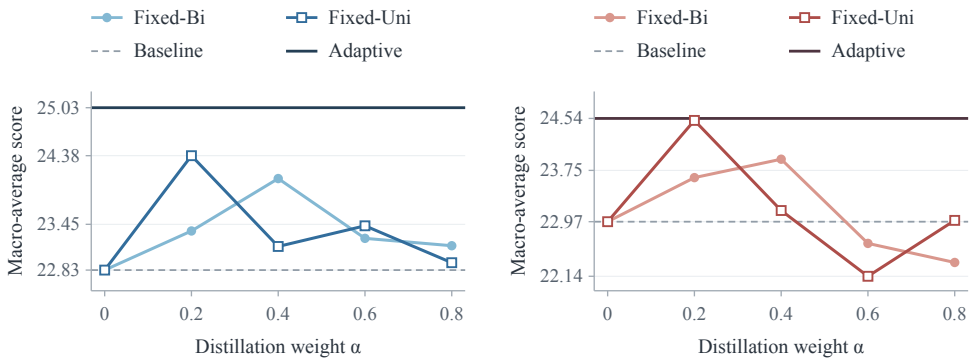
Table 8: Sampling-matched comparison on Llama-3.2-1B under temperature-smoothed sampling. Each reported model or branch uses the same $\tau > 1$ sampling distribution and six-benchmark evaluation protocol. All rows report raw scores; Δ Avg. gives the macro-average gain relative to the Smoothed SFT reference. Bold marks the highest value in each column.

Method	Sampling	Math		Medicine		Law		Avg.	Δ Avg.
		GSM8K	Math-500	MedQA	MedBullets	LawBench	CMMLU-Law		
SFT Baselines									
Smoothed SFT	$\tau > 1$	20.39	6.80	28.20	32.79	16.70	32.94	22.97	–
Prior Methods									
PCGrad	$\tau > 1$	21.30	7.00	26.47	34.90	20.78	33.14	23.93	+0.96
MFTCoder	$\tau > 1$	15.16	4.80	28.36	33.93	21.51	31.13	22.48	-0.49
CoBa	$\tau > 1$	16.91	6.00	25.69	33.44	21.89	33.08	22.84	-0.13
FAMO	$\tau > 1$	18.88	6.20	27.73	**36.36**	**23.86**	32.03	24.18	+1.21
Our Methods									
Fixed-Bi.	$\tau > 1$	**23.65**	8.80	**32.05**	30.19	17.25	31.57	23.92	+0.95
Fixed-Uni.	$\tau > 1$	20.55	**9.20**	29.77	33.60	23.82	30.11	24.51	+1.54
Adaptive	$\tau > 1$	22.44	7.80	26.87	34.09	21.76	**34.29**	**24.54**	**+1.57**

We rerun the prior single-model baselines listed in the table under the same temperature-smoothed sampling distribution ($\tau > 1$) as the reported AMD branch. Smoothed SFT serves as the sampling-matched reference.

Under this matched setting, FAMO is the strongest prior baseline at 24.18. The smoothed-sampling branch of Adaptive achieves the highest macro-average in the table at 24.54, improving over Smoothed SFT by 1.57 points and FAMO by 0.36 points. Fixed-Uni reaches 24.51, close to Adaptive. The advantage over the evaluated prior baselines therefore persists when their sampling distribution is matched to that of the reported AMD branch.

C Sensitivity to Fixed Distillation Weights



(a) Empirical sampling ($\tau = 1$) (b) Smoothed sampling ($\tau = 3$)

Figure 4: Sensitivity to fixed distillation weights on Llama-3.2-1B. Scores are six-benchmark macro-averages. Dashed and solid horizontal lines indicate the SFT baseline and Adaptive, respectively, for each sampling branch.

Figure 4 compares Adaptive with a sweep of fixed weights for Fixed-Bi and Fixed-Uni in the empirical- and smoothed-sampling branches on Llama-3.2-1B. Fixed-Uni achieves its highest plotted score at $\alpha = 0.2$ in both branches. For $\tau = 1$, Adaptive scores 25.03 versus the best fixed-weight result of 24.38, a gain of 0.65 points. For $\tau = 3$, Adaptive and Fixed-Uni score 24.54 and 24.51, respectively, a difference of 0.03 points. The gain from adaptation is therefore more pronounced in the empirical-sampling branch.

D WEIGHT-UPDATE SCHEDULE SENSITIVITY

Table 9: Sensitivity of the Llama-3.2-1B empirical-sampling model to the step-size schedule μ. Scores are six-benchmark macro-averages; bold marks the best score.

#	Schedule	Macro Avg.
1	$\mu = 1$	23.96
2	$\mu = \sqrt{\frac{T_{\max} - 0.8\,t}{T_{\max}}}$	24.89
3	$\mu = \sqrt{\max\left(0, \frac{T_{\max} - 1.2\,t}{T_{\max}}\right)}$	23.64
4	$\mu = \sqrt{\frac{T_{\max} - t}{T_{\max}}}$	**25.03**

Table 9 compares step-size schedules using the empirical-sampling branch. The standard square-root decay achieves 25.03, followed by slower decay at 24.89, a constant step at 23.96, and faster decay at 23.64. Both the standard and slower schedules outperform the constant and faster-decaying alternatives, supporting a gradual reduction in update magnitude.

E LONGER-BUDGET STRESS TEST

Table 10: Longer-budget stress test on Llama-3.2-1B. Single-model baselines train for six epochs, twice the three-epoch main protocol used by AMD. For each AMD regime, we report the higher-scoring model from the collaborative run. Scores are six-benchmark macro-averages.

Method	Macro Avg.
Single-model baselines, six epochs	
Multi-task SFT	23.05
Smoothed SFT	22.94
PCGrad	23.50
FAMO	24.16
CoBa	23.35
MFTCoder	23.52
HBO	24.11
AMD regimes, main protocol	
Fixed-Bi	24.07
Fixed-Uni	24.51
Adaptive	**25.03**

Table 10 evaluates single-model baselines trained for six epochs alongside the higher-scoring branch of each AMD regime trained under the three-epoch main protocol. FAMO achieves the highest single-model score of 24.16, compared with 24.07 for FIXED-BI, 24.51 for FIXED-UNI, and 25.03 for Adaptive. Adaptive retains the highest score when the single-model baselines receive twice as many training epochs. These epoch budgets correspond to different computational costs because AMD also uses cross-model forward passes and, in Adaptive, training probes and validation.

Adaptive performs five weight updates during training, each with three probes using 1% of a one-epoch training budget. The probes therefore add 0.15 epoch-equivalents of two-model training, or 5% of the regular three-epoch training budget. This estimate excludes validation and checkpoint-restoration overhead.