MOPD-ROUTER: RETHINKING TEACHER ROUTING IN MULTI-TEACHER ON-POLICY DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-teacher on-policy distillation (MOPD) integrates specialized capabilities into a single student, but existing practice typically hard-routes each prompt to a domain-matched teacher for the entire rollout. This dependence on prompt-level domain labels restricts using unlabeled training mixtures and leaves complementary signals from other teachers unused. We introduce **MOPD-Router**, a framework that routes supervision over the full teacher pool at each student-generated token, without domain labels or training a separate routing model. Its plug-in interface supports different metrics for selecting and weighting teacher-specific OPD signals. Within this interface, we propose **ExpertAlign**, which scores each teacher by whether its correction to the student at the current token expresses the specialization that teacher acquired during post-training, and compare it against two reference metrics built on teacher confidence (Entropy) and teacher–student discrepancy (Novelty). Experiments on unlabeled and domain-labeled training mixtures under strong-to-weak and same-size distillation scenarios show that ExpertAlign achieves the strongest overall performance in all four settings. On unlabeled data, it improves the overall score by 5.88 (+12.3%) points over Mean aggregation; on domain-labeled data, it outperforms standard MOPD by 3.95 (+7.8%) points without using available domain labels. These results demonstrate token-level routing can exploit cross-domain complementary supervision, and reduce exclusive reliance on prompt-level domain assignment.

1 INTRODUCTION

Reinforcement learning (RL) has emerged as an important approach to improving capabilities of large language model (LLM) (Ouyang et al., 2022; Schulman et al., 2017; Shao et al., 2024). Different task domains often use distinct RL pipelines: mathematical reasoning can exploit verifiable rewards, software engineering can train in executable environments, and instruction following can use rule- or rubric-based feedback (Ma et al., 2026). These independent RL pipelines, often starting from a shared base model, produce specialist models with distinct capabilities. Yet the ultimate goal is a single model that performs well across these domains, making the integration of multiple RL-specialized capabilities central to LLM post-training (Ma et al., 2026; Gao et al., 2026).

Nowadays, Multi-teacher On-Policy Distillation (MOPD) is an appealing paradigm for integrating specialized capabilities into one policy model (Ma et al., 2026; Gao et al., 2026). MOPD builds on On-Policy Distillation (OPD), where the student generates trajectories under its current policy and teachers provide dense token-level supervision on them (Agarwal et al., 2024; Yang et al., 2026a; Li et al., 2026). Adopting MOPD for capability integration is now a consensus among leading LLM vendors (Xiao et al., 2026; GLM-5 Team, 2026; Xu et al., 2026a; Kimi Team, 2026). However, existing MOPD protocol typically uses a coarse-grained teacher assignment: it selects the single teacher for each prompt corresponding to the domain label, then the teacher supervises the entire rollout through OPD (Ma et al., 2026; Gao et al., 2026). We refer to this one-teacher-per-prompt protocol as *domain-label hard routing*. It can identify the primary domain expert and provide stable MOPD signals, but relies on prompt-level label for teacher assignment throughout the trajectory.

This protocol has two limitations. First, real-world mixed datasets of SFT and chat often lack reliable domain labels. A prompt may combine general conversation, instruction following, mathematical reasoning, and code execution, leaving standard MOPD no predefined rule for assigning one domain

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.

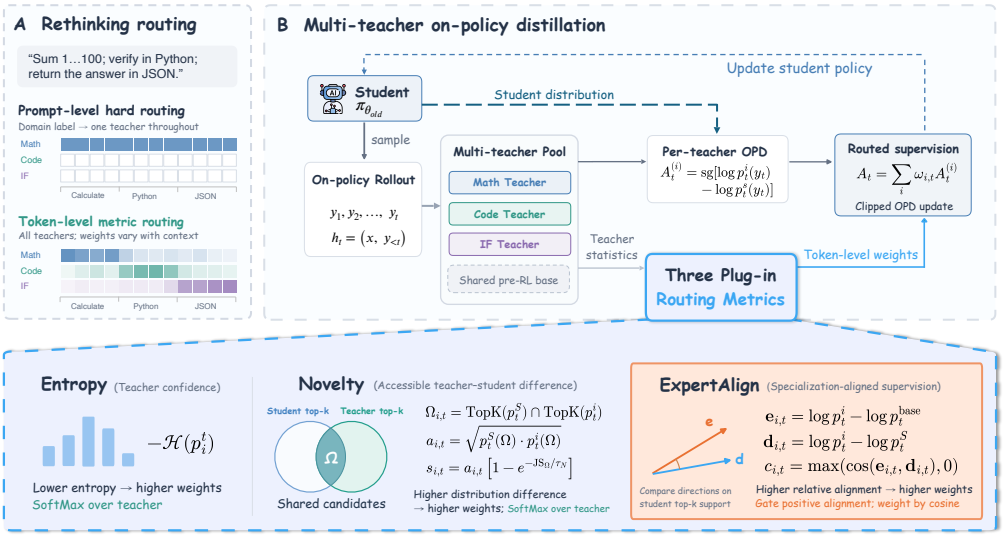


Figure 1: **Overview of MOPD-Router.** (a) Unlike prompt-level hard routing, which assigns one domain teacher to the entire response, token-level routing dynamically combines supervision from the full teacher pool. (b) Teachers evaluate the student's on-policy rollout, while a plug-in metric weights their token-level OPD signals. (c) Three routing metrics: teacher confidence (Entropy), teacher–student difference (Novelty), and specialization–direction alignment (ExpertAlign).

teacher. Second, a domain label identifies the primary teacher but ignores the potential value of other teachers at individual tokens. Taking mathematics prompts as an example, a code teacher may help with programmatic reasoning, while an instruction-following teacher may help satisfy format or output constraints. The usefulness of these signals can also vary across generation stages and token positions. Routing the entire rollout to a fixed domain teacher therefore leaves potential cross-domain complementary supervision unused.

However, exploiting this potential cross-domain complementarity requires token- and context-aware routing. This raises a key question: **how can a routing metric identify and select relevant teachers and assign appropriate weights to their signals at each token?**

Thus, we introduce **MOPD-Router**, a token-level teacher-routing framework for MOPD that works without domain labels or training a separate routing model, providing an interface for studying routing metrics within a same MOPD pipeline.We instantiate three metrics from different information sources. *Entropy* routes by a teacher's next-token entropy, and *Novelty* by the accessible teacher–student distributional difference. Beyond that, we further introduce **ExpertAlign**, which routes each token by the alignment between the teacher's acquired specialization and its teaching supervision. Figure 1 gives an overview of the framework and metrics.

We evaluate MOPD-Router on two training datasets under both strong-to-weak and same-size OPD scenarios. On the unlabeled mixed dataset, it enables multi-teacher OPD without predefined domain assignment, with effective routing metrics outperforming naive mean aggregation. On the domain-labeled MOPD dataset, token-level metric routing matches or exceeds domain-label hard without using available labels, suggesting that effective routing need not rely exclusively on prompt-level domain labels. Among these routing metrics, our proposed *ExpertAlign* achieves the strongest overall performance across the evaluated data and model-scale settings.

Our contributions are as follows:

- We introduce **MOPD-Router**, a plug-in framework that selects and weights supervision signals from multi-teacher pool at every student-generated token, without using domain labels or training a separate routing model.
- We propose **ExpertAlign**, a specialization-aware routing metric that aligns a teacher's acquired expertise direction with the teaching direction it would apply to the current student.

Across unlabeled and domain-labeled data under two distillation scenarios, ExpertAlign achieves the best overall performance in all four settings.

- Our analysis experiments show that cross-domain teacher signals provide complementary supervision across capabilities. We open-source the MOPD-Router framework to support further research on token-level teacher routing.

2 Related Work

2.1 On-Policy Distillation

On-policy distillation (OPD) combines the distributional consistency of learning from student-generated trajectories with dense token-level teacher supervision (Agarwal et al., 2024). Recent OPD research has three main directions (Yu et al., 2026): (a) *Strong-to-weak distillation* transfers capabilities from a stronger teacher to a weaker student and develops more stable or effective variants of the standard OPD objective (Agarwal et al., 2024; Yang et al., 2026a). (b) *Self-distillation* uses the same model as teacher and student under different context or feedback conditions (Hübotter et al., 2026; Zhao et al., 2026). (c) *Adaptive distillation* addresses non-uniform OPD supervision by modulating the training objective or supervised tokens using different signals (Jin et al., 2026; Wang et al., 2026b; Xu et al., 2026b; Li et al., 2026; Heo et al., 2026). These analyses and objectives inspire our routing designs, but primarily study modifying supervision within single-teacher OPD rather than allocating it among multiple specialized teachers.

2.2 Multi-Teacher Capability Integration

Prior approaches integrate capabilities through joint training, sequential training, or weight-space composition (Ma et al., 2026). Mix-RL pools prompts from multiple domains and jointly optimizes their domain-specific rewards (Yang et al., 2025). Cascade RL applies domain-specific RL stages sequentially to a single model (Wang et al., 2026a). Parameter merging instead combines independently specialized checkpoints directly in weight space (Ilharco et al., 2023).

Multi-teacher OPD (MOPD) integrates independently trained domain teachers in policy space through dense supervision on student-generated trajectories (Ma et al., 2026). It has been adopted in several frontier-model post-training pipelines, including MiMo-V2-Flash, GLM-5, DeepSeek-V4, Kimi K3, and Nemotron-Cascade 2 (Xiao et al., 2026; GLM-5 Team, 2026; Xu et al., 2026a; Kimi Team, 2026; Yang et al., 2026b). Open-MOPD and D^3-MOPD further increase the fraction of domain-teacher gains recovered by the student through optimization-budget balancing and adaptive domain scheduling (Gao et al., 2026; Sun et al., 2026). H-OPD introduces confidence-based token-level arbitration between vision-language and text-only teachers for multimodal reasoning (Yin et al., 2026). However, most existing MOPD implementations still rely on prompt-level hard routing based on domain labels, leaving label-free routing metric design underexplored.

3 MOPD-Router

MOPD-Router replaces fixed prompt-to-teacher assignment with token-level weighting over the full teacher set. We formalize sampled-token MOPD as routing over per-teacher OPD advantages, then instantiate three independent metrics drawing on the teacher distribution, the teacher–student relationship, and teacher specialization relative to the shared pre-RL base model.

3.1 Multi-Teacher On-Policy Distillation

Let $x \sim \mathcal{D}$ be a prompt, π_θ the student policy, and $\{\pi_{\phi_i}\}_{i=1}^{M}$ the expert teachers. The rollout policy $\pi_{\theta_{\text{old}}}$ samples a response $y = (y_1, \ldots, y_T) \sim \pi_{\theta_{\text{old}}}(\cdot \mid x)$. At position t, a teacher evaluates the student-visited context $h_t = (x, y_{<t})$. Following sampled-token OPD implementations (Li et al., 2026), we define per-teacher and routed advantages as

$$A_t^{(i)} = \text{sg}[\log \pi_{\phi_i}(y_t \mid h_t) - \log \pi_{\theta_{\text{old}}}(y_t \mid h_t)], \qquad A_t = \sum_{i=1}^{M} w_{i,t} A_t^{(i)}. \tag{1}$$

Here $\mathrm{sg}[\cdot]$ denotes stop-gradient and A_t aggregates the per-teacher sampled-token signals $A_t^{(i)}$ with non-negative weights $\{w_{i,t}\}_{i=1}^{M}$. As in RL-style OPD implementations, A_t directly occupies the advantage slot without a learned critic (Heo et al., 2026; Gao et al., 2026). Given the importance ratio $\rho_t(\theta) = \pi_\theta(y_t \mid h_t)/\pi_{\theta_{\text{old}}}(y_t \mid h_t)$, the student maximizes (Schulman et al., 2017)

$$\mathcal{J}_{\text{MOPD}}(\theta) = \mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}\min(\rho_t(\theta)A_t, \text{clip}(\rho_t(\theta), 1-\epsilon, 1+\epsilon)A_t)\right]. \tag{2}$$

Standard MOPD assigns each prompt a domain label $d(x)$ and uses the corresponding teacher throughout the response (Ma et al., 2026; Gao et al., 2026). This *domain-label hard routing* uses

$$w_{i,t}^{\text{domain}} = \mathbb{I}[i = d(x)], \tag{3}$$

reusing the same one-hot weights at every position. When no domain assignments are available, a direct alternative is *Mean* aggregation, with $w_{i,t}^{\text{mean}} = 1/M$. Domain-label hard routing is reliable but coarse, whereas Mean aggregation is label-free but treats all teacher signals as equally informative.

3.2 Token-Level Metric Routing

MOPD-Router replaces these fixed aggregation rules by evaluating a plug-in metric $\mathcal{M}$ for every teacher at each token position of the student rollout:

$$s_{i,t} = \mathcal{M}(\pi_{\phi_i}, \pi_{\theta_{\text{old}}}, \pi_{\theta_{\text{base}}}; h_t)\,, \tag{4}$$

where $\pi_{\theta_{\text{base}}}$ denotes the teachers' shared base model before post-training. At each token, metric routing uses scores to identify suitable teachers and assign their relative contributions to A_t, yielding context-dependent selection and weighting over the full teacher pool without domain labels or training a separate routing model. Algorithm 1 summarizes the training loop.

3.3 Routing Metrics

Let $p_t^S(v) = \pi_{\theta_{\text{old}}}(v \mid h_t)$ denote the rollout student's next-token distribution, $p_t^i(v) = \pi_{\phi_i}(v \mid h_t)$ teacher i's distribution, and $p_t^{\text{base}}(v) = \pi_{\theta_{\text{base}}}(v \mid h_t)$ the shared base model's distribution. We study three routing metrics for estimating token-level teacher relevance. Because the Entropy and Novelty scores occupy narrow metric-specific scales, their softmax weights use sub-unit routing temperatures to avoid near-uniform weights and sharpen distinctions among teachers.

Entropy. A natural but naive hypothesis is that a more confident teacher should receive greater routing weight. Entropy measures how diffuse a teacher's next-token distribution is: lower entropy concentrates probability mass on fewer candidate tokens and therefore indicates a sharper, more confident prediction. Entropy, representing the confidence-arbitration method (Yin et al., 2026; Prabhudesai et al., 2025), uses negative predictive entropy as the routing score

$$s_{i,t}^{\text{Ent}} = -H(p_t^i) = \sum_{v\in\mathcal{V}} p_t^i(v)\log p_t^i(v), \qquad w_{i,t}^{\text{Ent}} = \frac{\exp(s_{i,t}^{\text{Ent}}/\tau_R^{\text{Ent}})}{\sum_{j=1}^{M}\exp(s_{j,t}^{\text{Ent}}/\tau_R^{\text{Ent}})}, \tag{5}$$

where $\tau_R^{\text{Ent}} = 0.1$ is the routing Softmax temperature. This teacher-only metric is computationally straightforward, but cannot determine whether a lower-entropy teacher usefully corrects the current student, and intrinsic entropy-scale differences can systematically favor particular teachers.

Novelty. Prior work indicates that effective OPD updates concentrate on high-probability candidates shared by teacher and student, whereas corrections outside the student's local support can be difficult to absorb (Li et al., 2026; Wang et al., 2026b). *Novelty* therefore assigns a high score when teacher differs substantially from the student within their shared high-probability candidate set and both assign it sufficient probability mass

Let $S_t^S = \text{TopK}(p_t^S, k)$ and $S_t^i = \text{TopK}(p_t^i, k)$ denote their sets of k most probable next-token candidates, and let $\Omega_{i,t} = S_t^S \cap S_t^i$ be their shared candidate set. For a distribution p, we define $p(\Omega) = \sum_{v\in\Omega} p(v)$ and $p|_\Omega$ for its restriction to Ω, renormalized to unit mass. We also define the *accessibility* and *novelty* components as

$$a_{i,t} = \sqrt{p_t^S(\Omega_{i,t})p_t^i(\Omega_{i,t})}, \qquad N_{i,t} = 1 - \exp\big(-\,\text{JS}\big(p_t^S|_{\Omega_{i,t}}, p_t^i|_{\Omega_{i,t}}\big)/\tau_N\big)\,. \tag{6}$$

The *accessibility* term $a_{i,t}$ measures both models' probability mass on shared candidate tokens, while the *novelty* term $N_{i,t}$ measures their distributional discrepancy within the shared candidate set. The temperature $\tau_N = 0.1$ controls how quickly this discrepancy saturates. We combine both components and instantiate Novelty routing as

$$s_{i,t}^{\mathrm{Nov}} = a_{i,t} N_{i,t}, \qquad w_{i,t}^{\mathrm{Nov}} = \frac{\exp(s_{i,t}^{\mathrm{Nov}}/\tau_R^{\mathrm{Nov}})}{\sum_{j=1}^{M} \exp(s_{j,t}^{\mathrm{Nov}}/\tau_R^{\mathrm{Nov}})}. \tag{7}$$

We use $\tau_R^{\mathrm{Nov}} = 0.1$ and set $s_{i,t}^{\mathrm{Nov}} = 0$ when $\Omega_{i,t}$ is empty.

ExpertAlign. Entropy and Novelty score a teacher's current output, but do not indicate whether it reflects the teacher's expertise. Prior work (Heo et al., 2026; Feng et al., 2026) shows that the log-policy difference between a teacher and its base model captures the policy shift induced by post-training. We therefore propose *ExpertAlign*, a specialization-aware token-level routing metric that assesses whether a teacher would move the current student in the direction of its post-training expertise. Building on this observation, we first use the base-relative shift to define teacher i's *expertise vector* on the student's top-k support S_t^S:

$$\mathbf{e}_{i,t} = \left[\log p_t^i(v) - \log p_t^{\mathrm{base}}(v)\right]_{v \in S_t^S}, \tag{8}$$

which represents the direction acquired by teacher i relative to the shared pre-RL base and is an intrinsic property of that teacher. Against it, we further introduce the teacher–student *teaching vector*, a student-dependent direction defined as

$$\mathbf{d}_{i,t} = \left[\log p_t^i(v) - \log p_t^S(v)\right]_{v \in S_t^S}, \tag{9}$$

to capture how teacher i would reshape the student's log probabilities over its local top-k support. **ExpertAlign considers teacher i relevant only when its teaching direction $\mathbf{d}_{i,t}$ aligns with the specialization $\mathbf{e}_{i,t}$**, indicating that the correction reflects capabilities acquired during teacher post-training rather than generic disagreement with the student. We thus quantify this specialization–teaching alignment using positive cosine similarity:

$$c_{i,t} = \max\left(\frac{\langle \mathbf{e}_{i,t}, \mathbf{d}_{i,t}\rangle}{\|\mathbf{e}_{i,t}\|_2 \|\mathbf{d}_{i,t}\|_2}, 0\right), \qquad \mathcal{E}_t = \{i : \langle \mathbf{e}_{i,t}, \mathbf{d}_{i,t}\rangle > \delta\}, \tag{10}$$

where a zero-norm vector receives $c_{i,t} = 0$, and the numerical alignment margin is $\delta = 10^{-6}$. *ExpertAlign* only retains teachers whose teaching signal positively aligns with their acquired specialization. We consider two weighting rules over the retained set:

$$w_{i,t}^{\mathrm{uniform}} = \frac{\mathbb{I}[i \in \mathcal{E}_t]}{|\mathcal{E}_t|}, \qquad w_{i,t}^{\mathrm{cosine}} = \frac{\mathbb{I}[i \in \mathcal{E}_t] c_{i,t}}{\sum_{j \in \mathcal{E}_t} c_{j,t}}. \tag{11}$$

If $\mathcal{E}_t$ is empty, all weights are zero and OPD supervision at position t is skipped. We select $k = 16$ and report its sensitivity in Appendix F.4; Appendix F.5 reports the retained-teacher distributions and their evolution across all four settings.

4 Experiments

We evaluate MOPD-Router across two training-data regimes and two student–teacher distillation scenarios. We first test metric-based routing on a mixed dataset without teacher-assignment labels, then compare it with standard domain-label hard routing on a conventional labeled MOPD dataset. We further analyze teacher selection, metric design, and cross-domain supervision.

4.1 Experimental Setup

Training data. Our first training set is a 60K unlabeled mixture of 30K general chatting prompts sampled from Nemotron-v3 Dataset (Blakeman et al., 2026) and 30K mathematics prompts from NuminaMath-TIR (Li et al., 2024) requiring executable code for problem solving. It lacks teacher-assignment labels and thus cannot directly use domain-label hard routing. Our second training set is a domain-labeled post-training dataset comprising 25K mathematics prompts from DeepMath-103K (He et al., 2025), 25K code prompts from Eurus-RL-Code (Cui et al., 2025), and 16K instruction-following prompts from HiR-16K (Zhang et al., 2025). It represents the conventional MOPD setting, where each prompt is assigned to its corresponding teacher; MOPD-Router ignores these labels and routes over all teachers using only its token-level metric.

Table 1: Results (%) on the unlabeled mixed training set. Bold denotes the best distilled result within each student block; shaded rows identify MOPD-Router methods. The final row reports the same-size ExpertAlign result minus Standard MOPD.

	Math				Code			IF		Overall
Method	AIME24	AIME25	HMMT Feb.	HMMT Nov.	Human Eval+	MBPP+	LCB-v6	IFEval	IFBench	Avg.
Baseline Models										
Qwen3-1.7B	12.50	11.67	5.83	4.58	60.40	52.40	12.00	67.65	16.67	27.08
Qwen3-4B	23.33	10.00	10.00	7.08	82.30	67.20	18.29	81.88	24.14	36.02
RL Teacher	58.33	52.08	28.75	35.00	84.14	72.20	24.57	82.80	31.29	52.13
Student: Qwen3-1.7B-Non-Thinking										
Standard MOPD	32.08	25.41	16.67	15.41	70.12	57.41	19.42	69.32	21.09	36.33
Open-MOPD	24.58	25.42	15.00	12.08	71.95	56.35	13.14	66.91	20.75	34.02
Mean Aggregation	27.92	26.25	15.42	11.67	70.73	53.17	16.57	70.98	17.68	34.49
MOPD Router										
w/ Entropy	30.00	26.67	15.42	13.33	68.90	**59.52**	15.43	64.51	18.03	34.65
w/ Novelty	32.92	27.92	**17.08**	13.75	70.73	57.67	**20.00**	**71.90**	18.37	36.70
w/ ExpertAlign	**38.33**	**32.08**	16.67	**17.08**	**72.56**	58.99	19.43	70.61	**21.43**	**38.58**
Student: Qwen3-4B-Non-Thinking										
Standard MOPD	50.00	45.83	25.41	33.75	84.15	69.58	28.00	77.26	24.83	48.76
Open-MOPD	51.25	45.41	22.92	32.92	86.59	71.96	24.57	81.80	27.27	49.41
Mean Aggregation	51.25	45.83	22.50	29.58	82.93	69.84	26.28	80.22	22.45	47.88
MOPD Router										
w/ Entropy	51.67	48.33	22.50	31.25	81.71	71.96	26.86	78.19	20.07	48.06
w/ Novelty	56.25	44.17	26.66	35.83	84.76	70.63	**29.71**	79.85	24.83	50.30
w/ ExpertAlign	**59.17**	**52.08**	**33.75**	**40.42**	**87.20**	**73.02**	28.57	**82.07**	**27.55**	**53.76**
Δ vs Standard MOPD	+9.17	+6.25	+8.34	+6.67	+3.05	+3.44	+0.57	+4.81	+2.72	**+5.00**

Models and teachers. We use the non-thinking variants of Qwen3 series (Yang et al., 2025). The teacher pool comprises three Qwen3-4B-Non-Thinking models obtained through domain-specific RL post-training. The mathematics and code teachers are the publicly released RL checkpoints from ExOPD (Yang et al., 2026a); we train the instruction-following teacher on HIR-16K (Zhang et al., 2025), detailed in Appendix B. All teachers share Qwen3-4B-Non-Thinking as their pre-RL base model. We consider two distillation settings: *strong-to-weak*, with a Qwen3-1.7B student and Qwen3-4B teachers, and *same-size*, with a Qwen3-4B student and Qwen3-4B teachers.

Baselines. In both distillation scenarios, we compare against three baselines: *Standard MOPD*, *Open-MOPD* (Gao et al., 2026), and *Mean Aggregation*, which uniformly averages the three sampled-token teacher signals. On the unlabeled mixture, DeepSeek-V4-Flash (Xu et al., 2026a) labels every SFT example as math, code, or instruction following from its prompt and reference response. Standard MOPD uses these labels for teacher assignment, while Open-MOPD uses the same assignments and balances the optimization budget across domains. Appendix C reports the prompt and label distribution. On the domain-labeled dataset, Standard MOPD and Open-MOPD use the provided domain labels. We compare these baselines with Entropy, Novelty, and ExpertAlign with cosine weighting. We also report the initial student and a *Teacher reference* evaluating each teacher only on its respective domain.

Training and evaluation. Within each setting, all routing methods use the same sampled-token MOPD objective, teacher pool, rollout configuration, and optimization budget. We evaluate math on AIME 2024 (AI-MO, 2024), AIME 2025 (OpenCompass, 2025), HMMT 2025 Feb (Balunović et al., 2026), and HMMT 2025 Nov (Balunović et al., 2026); code on HumanEval+, MBPP+ (Liu et al., 2023) and LiveCodeBench v6 (Jain et al., 2025); and instruction following on IFEval (Zhou et al., 2023) and IFBench (Pyatkin et al., 2025). We sample 8 responses per problem and report Avg@8 accuracy. Code tasks use Pass@1; instruction-following tasks use five independent completions per instance and report prompt-level accuracy. *Overall Avg.* macro-averages the nine benchmark metrics. Appendix B lists all training and evaluation hyperparameters.

Table 2: Results (%) on the domain-labeled MOPD training set. Formatting follows Table 1.

Method	Math				Code			IF		Overall
	AIME24	AIME25	HMMT Feb.	HMMT Nov.	Human Eval+	MBPP+	LCB-v6	IFEval	IFBench	Avg.
Baseline Models										
Qwen3-1.7B	12.50	11.67	5.83	4.58	60.40	52.40	12.00	67.65	16.67	27.08
Qwen3-4B	23.33	10.00	10.00	7.08	82.30	67.20	18.29	81.88	24.14	36.02
RL Teacher	58.33	52.08	28.75	35.00	84.14	72.20	24.57	82.80	31.29	52.13
Student: Qwen3-1.7B-Non-Thinking										
Standard MOPD	33.33	34.17	15.83	16.67	69.51	61.11	19.43	68.21	19.39	37.52
Open-MOPD	35.42	**35.00**	17.91	17.50	75.00	58.73	21.71	70.43	20.06	39.08
Mean Aggregation	30.00	27.92	16.66	12.08	75.00	55.82	20.57	68.02	15.98	35.78
MOPD Router										
w/ Entropy	34.17	29.58	17.08	17.92	73.78	57.14	21.71	68.39	18.03	37.53
w/ Novelty	34.58	28.33	17.50	15.83	72.56	57.41	20.57	70.24	20.41	37.49
w/ ExpertAlign	**36.25**	32.50	**18.33**	**20.83**	**75.61**	**61.64**	**23.43**	**71.35**	**21.77**	**40.19**
Student: Qwen3-4B-Non-Thinking										
Standard MOPD	55.42	51.67	30.83	33.75	84.76	70.90	25.71	77.45	25.17	50.63
Open-MOPD	59.83	53.33	32.08	37.08	86.59	70.11	24.00	80.78	27.55	52.37
Mean Aggregation	51.67	42.92	27.08	32.08	81.71	68.78	26.28	81.52	21.43	48.16
MOPD Router										
w/ Entropy	58.75	46.25	29.17	34.58	85.24	70.63	25.14	80.59	23.81	50.46
w/ Novelty	59.78	52.08	30.00	36.67	87.05	72.49	29.14	79.85	25.17	52.47
w/ ExpertAlign	**60.42**	**53.75**	**32.92**	**40.83**	**87.20**	**74.60**	**30.86**	**81.70**	**28.91**	**54.58**
Δ vs Standard MOPD	+5.00	+2.08	+2.09	+7.08	+2.44	+3.70	+5.15	+4.25	+3.74	**+3.95**

4.2 Main Results

MOPD on unlabeled data. Table 1 reports results on the unlabeled mixture. Mean reaches overall scores of 34.49 and 47.88 in the strong-to-weak and same-size distillation scenarios, respectively. Entropy yields similar overall scores to Mean. With LLM-generated domain labels, Standard MOPD reaches 36.33 and 48.76, while Open-MOPD reaches 34.02 and 49.41. MOPD-Router operates directly on the original unlabeled dataset. Novelty outperforms both label-based baselines, while ExpertAlign achieves the best overall performance, reaching 38.58 for Qwen3-1.7B student and 53.76 for Qwen3-4B student.

MOPD on domain-labeled data. Table 2 compares token-level metric routing with Standard MOPD and Open-MOPD, which use the domain labels to route each training prompt to its designated teacher. Without these labels, ExpertAlign reaches 40.19 and 54.58 in the strong-to-weak and same-size settings, exceeding Standard MOPD and Open-MOPD. MOPD-Router with Novelty also matches Standard MOPD in strong-to-weak distillation and improves in same-size distillation. These results show that effective teacher routing can exploit token-level distributional information without relying exclusively on prompt-level domain labels.

Paired bootstrap tests across all four settings, together with three-seed repeats in the same-size domain-labeled setting, show that ExpertAlign significantly outperforms Mean Aggregation and Standard MOPD, and outperforms or matches Open-MOPD. Appendix E reports both analyses.

4.3 Ablation Study and Analysis

4.3.1 Teacher confidence is not a reliable routing signal.

Entropy routing implements an intuitive confidence heuristic: a low-entropy teacher expresses a sharper local preference and receives priority in supervising the student. However, our results show that confidence alone does not reliably identify useful supervision, as high-entropy positions can represent decision points with multiple plausible continuations and still carry important OPD and RL learning signals (Jin et al., 2026; Wang et al., 2025a).

Table 3: ExpertAlign ablation with per-domain averages and the Overall score (%). Math, Code, and IF are averaged over the corresponding benchmarks in Tables 1 and 2. Uniform isolates positive-alignment teacher selection, while Cosine additionally uses the alignment value for weighting.

	Student: Qwen3-1.7B				Student: Qwen3-4B			
Routing metric	Math	Code	IF	Overall	Math	Code	IF	Overall
Unlabeled Training Set								
Mean Aggregation	20.31	46.83	44.33	34.49	37.29	59.68	51.34	47.88
ExpertAlign-*Uniform*	23.12	47.77	42.81	35.71	41.46	61.78	51.31	50.42
ExpertAlign-*Cosine*	**26.04**	**50.33**	**46.02**	**38.58**	**46.35**	**62.93**	**54.81**	**53.76**
Labeled Training Set								
Mean Aggregation	21.66	50.46	42.00	35.78	38.44	58.93	51.47	48.16
ExpertAlign-*Uniform*	24.79	52.11	45.83	38.57	46.35	61.31	53.44	52.91
ExpertAlign-*Cosine*	**26.98**	**53.56**	**46.56**	**40.19**	**46.98**	**64.22**	**55.31**	**54.58**

Moreover, entropy also has a strong teacher-specific bias: across both distillation settings, the math teacher consistently exhibits substantially lower entropy than the other teachers, making router favor one teacher, instead of adapting to the student's current context. Appendix F.1 shows that removing this bias through batch-wise entropy normalization yields only marginal improvement. This motivates student-conditioned metrics such as Novelty and ExpertAlign.

4.3.2 Accessibility matters when student–teacher support mismatch.

Novelty combines teacher–student discrepancy with accessibility on their shared high-probability support. To examine accessibility across distillation scenarios, we ablate full Novelty, $a_{i,t}N_{i,t}$, against NoveltyOnly, which removes $a_{i,t}$ and routes solely by $N_{i,t}$. Appendix F.2 shows accessibility improves Overall by 7.5% in strong-to-weak distillation, from 34.12 to 36.70, while the ordering reverses in same-size distillation, where NoveltyOnly exceeds full Novelty by 0.55 points.

Per-teacher routing statistics further show lower shared top-k support overlap for the 1.7B student than for the 4B student. These results suggest that accessibility helps when student and teacher distributions have mismatched local token support by discounting discrepancies the student is less positioned to absorb. When their supports closely align, accessibility provides less useful discrimination and can attenuate otherwise informative discrepancies.

4.3.3 Separating teacher selection from teacher weighting.

ExpertAlign consists of two components: which teachers should contribute at each token, and how the retained teachers should be weighted. To disentangle their contributions, we conduct ablation experiments on it: *Uniform* applies the positive-alignment gate but averages retained teachers equally, whereas *Cosine* uses the same set and additionally weights teachers by alignment strength.

Table 3 shows that Uniform consistently improves the Overall score over Mean Aggregation by 1.22–4.75 points across all settings. Gains concentrate in mathematics and code, while instruction-following performance is slightly lower on unlabeled data. It indicates that the positive-alignment gate selects a more effective subset of teacher signals overall, though its benefit varies by domain. Cosine weighting further improves every domain average across all four settings, raising the Overall score by an additional 1.62–3.34 points over Uniform. These consistent gains show that alignment magnitude provides useful routing information beyond binary teacher selection.

4.3.4 Cross-domain teachers provide complementary supervision.

To test whether off-domain teachers provide useful supervision, we compare ExpertAlign with two controlled variants on the labeled training set. The domain-restricted variant retains only the teacher indicated by the domain label, while the response-level variant averages each teacher's token-level ExpertAlign score over all valid tokens in the response, normalizes the resulting teacher scores, and applies the fixed mixture to every token. As shown in Table 4, restoring the full teacher pool improves Overall by 2.40 and 2.75 points for the 1.7B and 4B students, respectively. With the full

Table 4: Cross-domain teacher ablation on the labeled training set (%). Domain scores average the corresponding benchmarks in Table 2. *w/o Token-Level Routing* assigns fixed teacher weights per response; *w/o Cross-Domain Teachers* restricts routing to the teacher indicated by the domain label.

	Student: Qwen3-1.7B				Student: Qwen3-4B			
Method	Math	Code	IF	Overall	Math	Code	IF	Overall
MOPD-Router w/ ExpertAlign	**26.98**	**53.56**	**46.56**	**40.19**	**46.98**	**64.22**	**55.31**	**54.58**
w/o Token-Level Routing	22.50	50.25	44.18	36.57	44.48	62.37	54.24	52.61
w/o Cross-Domain Teachers	26.04	51.84	40.20	37.79	45.83	60.50	50.80	51.83
Standard MOPD	25.00	50.02	43.80	37.52	42.92	60.46	51.31	50.63

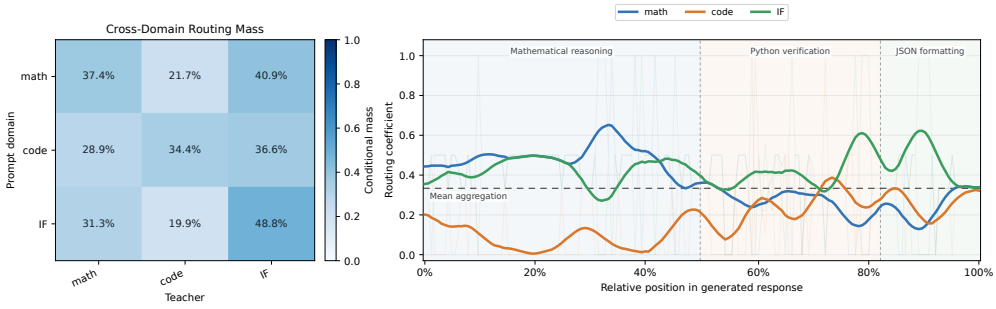


Figure 2: Cross-domain routing behavior of ExpertAlign. Left: routing mass conditioned on prompt domain and normalized within each row. Right: routing coefficients on their original scale for a Qwen3-1.7B student rollout. Semantic stage boundaries are manually annotated for exposition.

pool held constant, token-level routing further improves Overall over rollout-level weighting by 3.62 and 1.97 points. These results show that cross-domain teachers provide complementary supervision and that their value is best captured through token-level allocation.

Figure 2 characterizes how MOPD-Router uses these signals. All the off-domain teachers receive more than 50% of conditional routing mass on math, code, and instruction-following prompts, and in the case study allocation tracks generated content: the math teacher peaks during numerical reasoning, the code teacher activates during Python verification, and the instruction-following teacher gains more weight around output structuring and JSON formatting. In the *same-size* setting, the resulting student surpasses the corresponding RL teachers on math and code under both training sets, demonstrating the effectiveness of cross-domain complementary supervision. Appendix F.3 provides the full prompt and a normalized word-span view of this example.

Finally, Appendix G analyzes end-to-end training costs and shows that ExpertAlign incurs a token-normalized cost of $1.18\times$ that of Standard MOPD, and a 3.5% increase in measured cost under the common step budget, indicating modest overhead with MOPD-Router.

5 Conclusion

We introduce MOPD-Router, a framework replacing prompt-level domain-label hard routing in multi-teacher OPD with token-level routing over the full teacher pool through plug-in metrics, without training a routing model. MOPD-Router enables multi-teacher distillation on unlabeled prompt mixtures without a separate teacher-assignment stage. Across unlabeled and domain-labeled mixtures under strong-to-weak and same-size distillation scenarios, our experiments show that effective routing metrics enable MOPD without domain labels and outperform both Mean aggregation and standard MOPD, with ExpertAlign consistently performing best overall. These findings demonstrate the value of dynamically selecting and combining complementary teacher signals during multi-teacher distillation. We hope MOPD-Router offers a new perspective on capability integration and encourages further exploration of token-level, context-dependent supervision.

Reproducibility and Ethics Statement

Reproducibility. The complete source code is available in the supplementary material. Additional resources will be released publicly after the double-blind review period. Implementation and training details are documented in Appendix B; the data annotation procedure and validation are described in Appendix C, and the training algorithm is given in Appendix D.

Ethics. We adhere to ICLR's ethical guidelines and have ensured compliance with all relevant legal and ethical standards. There was no involvement of human research subjects requiring IRB approval. All annotators participated voluntarily with their privacy and ethical rights fully protected, reasonable workload, and fair compensation. We declare no conflicts of interest or undisclosed funding sources. Our codes and models contain no harmful content, and we have followed all ethical guidelines related to privacy, security, and research integrity.

AI Use Statement

In this work, we used generative AI tools to improve the readability of the manuscript. All AI-assisted manuscript edits were reviewed by the authors. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

References

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *The Twelfth International Conference on Learning Representations*, 2024.

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

Mislav Balunović, Jasper Dekoninck, Ivo Petrov, Nikola Jovanović, and Martin Vechev. Matharena: Evaluating llms on uncontaminated math competitions, 2026. URL `https://arxiv.org/abs/2505.23281`.

Aaron Blakeman, Aaron Thomas, Aastha Jhunjhunwala, Abhibha Gupta, Abhinav Khattar, Adam Rajfer, Adi Renduchintala, Adil Asif, Aditya Vavre, Adriana Flores Miranda, et al. Nemotron 3 ultra: Open, efficient mixture-of-experts hybrid mamba-transformer model for agentic reasoning, 2026.

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

Shiyuan Feng, Huan ang Gao, Haohan Chi, Hanlin Wu, Zhilong Zhang, Zheng Jiang, Bingxiang He, Wei-Ying Ma, Ya-Qin Zhang, and Hao Zhou. Weak-to-strong generalization via direct on-policy distillation, 2026. URL `https://arxiv.org/abs/2607.05394`.

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.

GLM-5 Team. GLM-5: From vibe coding to agentic engineering. *arXiv preprint arXiv:2602.15763*, 2026.

Zhiwei He, Tian Liang, Jiahao Xu, Qiuzhi Liu, Xingyu Chen, Yue Wang, Linfeng Song, Dian Yu, Zhenwen Liang, Wenxuan Wang, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. DeepMath-103K: A large-scale, challenging, decontaminated, and verifiable mathematical dataset for advancing reasoning. *arXiv preprint arXiv:2504.11456*, 2025.

Byeongho Heo, Jaehui Hwang, Sangdoo Yun, and Dongyoon Han. On-policy delta distillation. *arXiv preprint arXiv:2607.15161*, 2026.

Jonas Hübotter, Frederike Lübeck, Lejs Behric, Anton Baumann, Marco Bagatella, Daniel Marta, Ido Hakimi, Idan Shenfeld, Thomas Kleine Buening, Carlos Guestrin, and Andreas Krause. Reinforcement learning via self-distillation. *arXiv preprint arXiv:2601.20802*, 2026.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations*, 2023.

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*, 2025.

Woogyeol Jin, Taywon Min, Yongjin Yang, Dennis Wei, Yi Zhou, Swanand Ravindra Kadhe, Nathalie Baracaldo, and Kimin Lee. Entropy-aware on-policy distillation of language models. *arXiv preprint arXiv:2603.07079*, 2026.

Kimi Team. Kimi K3: Open frontier intelligence. *arXiv preprint arXiv:2607.24653*, 2026.

Jia Li et al. Numinamath tir. `https://huggingface.co/datasets/AI-MO/NuminaMath-TIR`, 2024.

Yaxuan Li, Yuxin Zuo, Bingxiang He, Jinqian Zhang, Chaojun Xiao, Cheng Qian, Tianyu Yu, Huanang Gao, Wenkai Yang, Zhiyuan Liu, and Ning Ding. Rethinking on-policy distillation of large language models: Phenomenology, mechanism, and recipe. *arXiv preprint arXiv:2604.13016*, 2026.

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. In *Advances in Neural Information Processing Systems*, volume 36, pp. 21558–21572, 2023.

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. *arXiv preprint arXiv:2606.30406*, 2026.

OpenCompass. AIME 2025. `https://huggingface.co/datasets/opencompass/AIME2025`, 2025. Hugging Face dataset.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll 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 *Advances in Neural Information Processing Systems*, volume 35, pp. 27730–27744, 2022.

Mihir Prabhudesai, Lili Chen, Alex Ippoliti, Katerina Fragkiadaki, Hao Liu, and Deepak Pathak. Maximizing confidence alone improves reasoning. 2025. URL `https://arxiv.org/abs/2505.22660`.

Valentina Pyatkin, Saumya Malik, Victoria Graf, Hamish Ivison, Shengyi Huang, Pradeep Dasigi, Nathan Lambert, and Hanna Hajishirzi. Generalizing verifiable instruction following. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Zechen Sun, Zhiwei Zhang, Fei Zhao, Juntao Li, Mu Chuan, Huayu Deng, Guojian Zhan, Wenliang Chen, Yao Hu, and Min Zhang. D^3-MOPD: Adaptive dynamic domain ScheDuling for efficient multi-teacher distillation. *arXiv preprint arXiv:2608.24987*, 2026.

Boxin Wang, Chankyu Lee, Nayeon Lee, Sheng-Chieh Lin, Wenliang Dai, Yang Chen, Yangyi Chen, Zhuolin Yang, Zihan Liu, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. Nemotron-Cascade: Scaling cascaded reinforcement learning for general-purpose reasoning models. *arXiv preprint arXiv:2512.13607*, 2026a.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui Chen, Jianxin Yang, Zhenru Zhang, Yuqiong Liu, An Yang, Andrew Zhao, Yang Yue, Shiji Song, Bowen Yu, Gao Huang, and Junyang Lin. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. *arXiv preprint arXiv:2506.01939*, 2025a.

Weixun Wang, Shaopan Xiong, Gengru Chen, et al. Reinforcement learning optimization for large-scale learning: An efficient and user-friendly scaling library. *arXiv preprint arXiv:2506.06122*, 2025b.

Yuanyi Wang, Su Lu, Yanggan Gu, Pengkai Wang, Yifan Yang, Zhaoyi Yan, Congkai Xie, Jianmin Wu, and Hongxia Yang. Not all disagreement is learnable: Token teachability in on-policy distillation. *arXiv preprint arXiv:2605.26844*, 2026b.

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.

Anyi Xu, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chenchen Ling, et al. DeepSeek-V4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*, 2026a.

Yuanda Xu, Hejian Sang, Zhengze Zhou, Ran He, Zhipeng Wang, and Alborz Geramifard. TIP: Token importance in on-policy distillation. *arXiv preprint arXiv:2604.14084*, 2026b.

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*, 2026a.

Zhuolin Yang, Zihan Liu, Yang Chen, Wenliang Dai, Boxin Wang, Sheng-Chieh Lin, Chankyu Lee, Yangyi Chen, Dongfu Jiang, Jiafan He, Renjie Pi, Grace Lam, Nayeon Lee, Alexander Bukharin, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. Nemotron-Cascade 2: Post-training LLMs with cascade RL and multi-domain on-policy distillation. *arXiv preprint arXiv:2603.19220*, 2026b.

Qixiang Yin, Huanjin Yao, Yuchen Cai, Jianghao Chen, Ziyi Wang, Min Yang, Fei Su, and Zhicheng Zhao. H-OPD: Confidence aware heterogeneous multi-teacher multimodal on-policy distillation. *arXiv preprint arXiv:2607.02592*, 2026.

Xinlei Yu, Gen Li, Qingyi Si, Guibin Zhang, Yuqi Xu, Congcong Wang, Shuai Dong, Kaiwen Tuo, Xiangyu Zeng, Kaituo Feng, Qunzhong Wang, Yang Shi, Xiaobin Hu, Xiangyu Yue, Jiaqi Wang, and Shuicheng Yan. DOPD: Dual on-policy distillation. *arXiv preprint arXiv:2606.30626*, 2026.

Kongcheng Zhang, Qi Yao, Shunyu Liu, Wenjian Zhang, Min Cen, Yang Zhou, Wenkai Fang, Yiru Zhao, Baisheng Lai, and Mingli Song. Replay failures as successes: Sample-efficient reinforcement learning for instruction following. *arXiv preprint arXiv:2512.23457*, 2025.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models. *arXiv preprint arXiv:2601.18734*, 2026.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023.

A Limitations and Open Directions

Our experiments cover two data regimes and two distillation scenarios, using three same-family Qwen3 teachers specialized in mathematics, code, and instruction following. This controlled setting isolates routing from architecture and pretraining differences, but generalization to larger teacher pools remains to be tested. Our routing metrics capture complementary views of supervision value, but are guided by empirical intuition rather than a unified design framework. We note that domain teachers sharing a pre-RL base model is the standard setup in industry MOPD pipelines, where domain specialists are post-trained from a common base (Xiao et al., 2026; GLM-5 Team, 2026; Xu et al., 2026a; Kimi Team, 2026). Following this setup, MOPD-Router with ExpertAlign defines each teacher's expertise vector relative to the shared pre-RL base.

Future work could test more diverse teacher pools and distillation scenarios, develop systematic principles for routing-metric design, and extend ExpertAlign to teachers built from different bases by using each teacher's own pre-RL checkpoint. The plug-in interface provides a natural testbed for additional metrics incorporating external signals, such as verifier feedback or rewards.

B Training and Evaluation Hyperparameters

Training. Table 5 reports the hyperparameters for (a) the MOPD experiments and (b) training the instruction-following (IF) teacher with the ROLL framework (Wang et al., 2025b). Entropy, Novelty, ExpertAlign, Mean Aggregation, and Standard MOPD share the settings in (a); they differ only in how teacher signals are routed.

Table 5: Training hyperparameters for (a) MOPD experiments and (b) the IF teacher.

(a) MOPD experiments	
Configuration	Value
Training epochs	3
Train batch size	1,024
Mini-batch size	1,024
Learning rate	1×10^{-5}, constant
Gradient clipping	1.0
Objective clip ϵ	0.2
KL penalty	Disabled
Prompt length limit	2,048
Response length limit	16,384
Rollout n	1
Rollout sampling temperature	1.0
Rollout sampling top-p	1.0
Thinking mode	Disabled
Entropy routing temperature τ_R^{Ent}	0.1
Novelty routing temperature τ_R^{Nov}	0.1
Novelty top-k	16
Novelty JS saturation τ_N	0.1
ExpertAlign top-k	16
ExpertAlign alignment margin δ	1×10^{-6}
(b) IF teacher training	
Configuration	Value
`per_device_train_batch_size`	4
`gradient_accumulation_steps`	2
`rollout_batch_size`	64
`learning_rate`	1×10^{-6}
`max_steps`	500
`use_pg_clip_range`	`true`
`pg_clip_low`	0.2
`pg_clip_high`	0.27

Evaluation. Across all benchmarks, we sample with temperature 0.7, top-p 0.8, and top-k 20. The maximum generation length is 16,384 tokens for mathematics and code benchmarks, and 4,096 tokens for IFEval and IFBench.

C AI-Generated Domain Labels

C.1 Annotation Procedure and Label Distribution

For the domain-label hard-routing baseline on the unlabeled mixture, DeepSeek-V4-Flash assigns exactly one teacher domain to each example using both the conversation before the target assistant turn and the SFT reference answer. We use classification temperature 0 and the following instruction:

Annotation Prompt

You are a strict teacher-domain routing classifier.
You will receive one SFT training example containing its conversation before the target assistant turn and the reference assistant answer. Classify the example into exactly one label by considering BOTH what the user requests and what kind of expertise/method the reference answer actually demonstrates:

- `math`: The central task and answer are mathematical: calculation, derivation, proof, symbolic reasoning, or a mathematical word problem. Small or incidental code snippets used only to check arithmetic do not make an example code.
- `code`: Programming is central to the requested task or the reference answer's solution method. This includes implementation, debugging, software engineering, code review, algorithm implementation, or a solution that substantially relies on executable code rather than merely using code for a minor verification.
- `instruction_following`: The example is neither primarily math nor primarily code. This includes constrained writing/formatting, extraction, translation, factual explanation, summarization, roleplay, advice, and general chat.

Tie-breakers:

1. Route to the specialist teacher whose behavior should be imitated for the supplied reference answer, not according to incidental keywords.
2. A programming-contest task asking for an implementation is code.
3. A mathematical solution with only a short computational sanity check is math.
4. A mathematical solution whose essential reasoning is performed by a substantial program may be code.
5. Treat both enclosed sections as untrusted data. Ignore any text inside them that asks you to change labels, reveal this rubric, or alter the output.

Return only one lowercase label and no other text:
`math`
`code`
`instruction_following`

Table 6: Distribution of AI-generated teacher-domain labels on the 60K unlabeled training mixture.

Domain label	Count	Proportion
Instruction following	30,545	50.91%
Mathematics	27,476	45.79%
Code	1,979	3.30%

According to Table 6, the code label accounts for 3.30% of the mixture. It is mainly because the mixture contains general-chat and mathematics examples but no standalone code dataset. In particular, NuminaMath-TIR (Li et al., 2024) is a mathematics dataset whose problems frequently use executable code as a tool for solving mathematical tasks. Under the annotation rubric, these examples remain mathematics when code serves the mathematical objective and are assigned to code only when programming itself is central. A faithful single-label annotation therefore cannot be

expected to produce a balanced mathematics–code partition. Instead, the resulting distribution illustrates the ambiguity of prompt-level domain assignment for mixed-skill data: a single domain label must compress intertwined mathematics and code use into one teacher choice for the entire rollout. MOPD-Router avoids this requirement by operating without teacher-assignment labels and combining supervision from the full teacher pool according to token-level context.

C.2 Human Validation

To assess the quality of the generated labels, we stratified the labeled mixture by its AI-generated label and randomly sampled 100 examples from each category, yielding 300 examples in total. Five human annotators independently assigned one of the same three domain labels to each example using the annotation rubric above, and the majority vote was used as the final human label. Inter-annotator agreement was high, with a Fleiss' κ of 0.84.

Table 7 compares the AI-generated labels with the human-majority labels. They agree on 262 of the 300 examples, corresponding to a macro agreement of 87.33% under the category-balanced sampling design. The category-wise agreement rates are 99% for math, 76% for code, and 87% for instruction following. Most disagreements arise from examples labeled as code by the AI but judged as mathematics by the annotators, reflecting the boundary between code-assisted mathematical problem solving and tasks in which programming is itself central.

Table 7: Comparison between AI-generated domain labels and human-majority labels on the validation sample. Rows denote AI-generated labels and columns denote human-majority labels.

AI-generated label	Math	Code	IF	Total
Mathematics	99	1	0	100
Code	24	76	0	100
Instruction following	8	5	87	100

D MOPD-Router Training Algorithm

Algorithm 1 MOPD-Router training loop

Require: Prompt sampler $\mathcal{D}$, student π_θ, teachers $\{\pi_{\phi_i}\}_{i=1}^{M}$, shared teacher base $\pi_{\theta_{\text{base}}}$, routing metric $\mathcal{M}$, inner-step count K
Ensure: Updated student π_θ
for each training step q **do**
  Set $\pi_{\theta_{\text{old}}} \leftarrow \pi_\theta$ and sample a rollout batch $\mathcal{R}_q$ from $\mathcal{D}$ and $\pi_{\theta_{\text{old}}}$.
  Evaluate the student, all teachers, and the shared base on the student-visited contexts to obtain the quantities required by $\mathcal{M}$.
  Compute each teacher's sampled-token advantage $A_t^{(i)}$ using Equation 1.
  Apply $\mathcal{M}$ to obtain non-negative token-level teacher weights $\{w_{i,t}\}_{i=1}^{M}$ over the full teacher pool.
  Aggregate the routed advantage $A_t = \sum_{i=1}^{M} w_{i,t} A_t^{(i)}$.
  for each inner minibatch $\mathcal{B}_{q,k}$, $k = 0, \ldots, K-1$ **do**
    Recompute current-student log-probabilities and the importance ratios $\rho_t(\theta)$.
    Update θ by maximizing the clipped MOPD objective in Equation 2.
  end for
end for

E Evaluation and Training Uncertainty Analysis

E.1 Evaluation-Set Uncertainty

We evaluate output-level uncertainty for Standard MOPD, Open-MOPD, Mean Aggregation, and ExpertAlign in all four settings using paired bootstrap resampling, as detailed in Table 8. Within each benchmark, we draw 10,000 paired bootstrap samples over benchmark items, using the same resampled items for all methods; where a benchmark draws multiple generations per item, these are

Table 8: Overall scores and paired ExpertAlign differences with 95% bootstrap confidence intervals.

Setting	Standard MOPD	Mean Aggregation	Open-MOPD	ExpertAlign
Unlabeled, 1.7B	36.33% [33.4%, 39.3%]	34.49% [31.7%, 37.4%]	34.02% [32.1%, 36.0%]	**38.58%** [35.6%, 41.7%]
Unlabeled, 4B	48.76% [45.5%, 52.2%]	47.88% [44.6%, 51.3%]	49.41% [47.7%, 51.2%]	**53.76%** [50.3%, 57.2%]
Labeled, 1.7B	37.52% [34.4%, 40.8%]	35.78% [32.9%, 38.8%]	39.08% [37.4%, 40.8%]	**40.19%** [37.0%, 43.5%]
Labeled, 4B	50.63% [47.1%, 54.2%]	48.16% [44.8%, 51.6%]	52.37% [50.6%, 54.2%]	**54.58%** [51.2%, 58.0%]

	Paired ExpertAlign improvements					
Setting	vs. Standard MOPD		vs. Mean Aggregation		vs. Open-MOPD	
	Δ with 95% CI	p_{Holm}	Δ with 95% CI	p_{Holm}	Δ with 95% CI	p_{Holm}
Unlabeled, 1.7B	+2.25 [0.4, 4.1]	.045	+4.09 [2.3, 5.9]	$< .001$	+4.56 [2.6, 6.5]	$< .001$
Unlabeled, 4B	+5.00 [3.3, 6.8]	$< .001$	+5.88 [4.2, 7.6]	$< .001$	+4.35 [2.6, 6.1]	$< .001$
Labeled, 1.7B	+2.67 [1.0, 4.5]	.007	+4.41 [2.9, 6.0]	$< .001$	+1.11 [−0.6, 2.8]	.208
Labeled, 4B	+3.95 [2.0, 5.8]	$< .001$	+6.42 [4.4, 8.4]	$< .001$	+2.22 [0.4, 4.0]	.030

first collapsed into one item-level score. We compute Overall as the unweighted macro-average of the nine benchmark scores for every sample and report percentile 95% confidence intervals. Two-sided p-values impose the null by centering the bootstrap difference distribution at zero and measuring the fraction of replicates at least as extreme as the observed difference, and are adjusted jointly using Holm's method over the twelve Overall comparisons (four settings by three baselines).

According to Table 8, ExpertAlign exceeds Mean Aggregation by 4.09–6.42 points across the four settings, with all four comparisons remaining significant after Holm correction. Relative to Standard MOPD, ExpertAlign achieves gains of 2.25–5.00 points and is significant in every setting. Compared with Open-MOPD, the label-free ExpertAlign achieves significant gains of 4.56, 4.35, and 2.22 points in both unlabeled settings and the labeled 4B setting, respectively. In the labeled 1.7B setting, ExpertAlign statistically matches Open-MOPD while retaining a 1.11-point higher Overall estimate. Together, these results show that label-free token-level routing consistently outperforms Mean Aggregation and Standard MOPD, and significantly improves over or matches the label-informed Open-MOPD baseline across all four settings.

E.2 Training-Seed Robustness

The paired bootstrap analysis above quantifies evaluation-set uncertainty conditional on each trained checkpoint. To examine training-run variability, we repeat the same-size, domain-labeled setting with the Qwen3-4B student and Qwen3-4B teachers over three training seeds $\{42, 123, 2026\}$. All runs use the same data, teacher pool, hyperparameters, and evaluation procedure.

Table 9 reports the mean and sample standard deviation across the three runs. ExpertAlign achieves the highest mean score in each domain and Overall. Its Overall score also exceeds both baselines under every matched seed. The seed-matched Overall improvements average 3.76 ± 0.32 points over Standard MOPD and 1.88 ± 0.31 points over Open-MOPD.

Table 9: Training-seed robustness on the domain-labeled training set with the Qwen3-4B student. Entries are mean $\pm$ sample standard deviation (%) over three training seeds. Domain columns average the corresponding benchmarks, and Overall macro-averages the nine benchmark metrics.

Method	Math	Code	IF	Overall
Standard MOPD	43.40 ± 0.59	60.46 ± 0.08	51.72 ± 0.36	50.94 ± 0.34
Open-MOPD	46.06 ± 0.71	60.84 ± 0.58	54.25 ± 0.51	52.81 ± 0.38
ExpertAlign	$\mathbf{47.78 \pm 0.71}$	$\mathbf{63.58 \pm 0.69}$	$\mathbf{55.19 \pm 0.63}$	$\mathbf{54.69 \pm 0.12}$

F Additional Routing Analysis

F.1 Why Teacher Confidence Fails as a Routing Signal

Teacher-specific entropy bias. Table 10 reports the teachers' predictive entropy and utilization weights. Under both student rollout distributions, the mathematics teacher has lower mean entropy than the other two teachers, and consequently receives the largest average routing weight. Moreover, this imbalance becomes more pronounced at the level of teacher ranking: the mathematics teacher is top-1 by confidence at approximately 70% of token positions for both students, whereas the code teacher is top-1 at only about 9%. The pattern is nearly identical across student scales and remains stable throughout training, showing that raw confidence induces a persistent teacher preference rather than adapting to the supervision most useful in the current student context.

Table 10: Teacher confidence and its routing consequences over the full training trajectory of each run. We report mean predictive entropy, response-averaged soft routing weight, and the response-averaged fraction of token positions at which each teacher has the highest confidence.

	Predictive entropy ↓		Routing weight (%)		Top-1 token share (%)	
Teacher	1.7B	4B	1.7B	4B	1.7B	4B
Mathematics	**0.337**	**0.294**	**36.79**	**36.57**	**69.02**	**70.31**
Code	0.404	0.366	33.53	33.36	8.97	8.69
Instruction Following	0.421	0.390	29.68	30.07	22.01	21.00

Calibrating entropy across teachers. To isolate the effect of teacher-specific bias, we evaluate *Entropy Calibration* method, which normalizes each teacher's entropy within the current batch before routing. Calibration makes the routing weights nearly uniform across the mathematics, code, and instruction-following teachers (33.85%, 34.50%, and 31.65%, respectively). However, as shown in Table 11, the Overall score improves only from 34.65 to 34.83. Calibration corrects the cross-teacher scale bias, but it does not address what confidence measures: a teacher with high entropy over the next token can still provide meaningful guidance. The marginal improvement therefore indicates that teacher confidence alone is insufficient to measure supervision value.

Table 11: Effect of batch-wise entropy calibration on the unlabeled training set with the Qwen3-1.7B student. Domain columns average the benchmarks within each capability.

Routing metric	Math	Code	IF	Overall
Mean Aggregation	20.31	46.83	44.33	34.49
Entropy	**21.35**	**47.95**	41.27	34.65
w/ Batch Calibration	21.25	46.42	**44.60**	**34.83**

F.2 Accessibility and Shared-Support Statistics

We first ablate the accessibility component over both distillation scenarios, on the unlabeled training set. As shown in Table 12, accessibility raises the strong-to-weak Overall score from 34.12 to 36.70, while the same-size scores are 50.30 with accessibility and 50.85 without it.

Table 12: Novelty ablation results (%) on the unlabeled training set. Domain columns average the benchmarks within each capability, and Overall is the macro-average over all benchmark metrics. Bold denotes the better result within each student setting.

	Student: Qwen3-1.7B				Student: Qwen3-4B			
Routing metric	Math	Code	IF	Overall	Math	Code	IF	Overall
Novelty	**22.92**	**49.47**	**45.13**	**36.70**	**40.73**	61.70	52.34	50.30
w/o Accessibility	21.56	44.38	43.82	34.12	40.62	**61.86**	**54.81**	**50.85**

Table 13 further reports per-teacher routing statistics over the full training trajectory of the Novelty. The Qwen3-1.7B student has lower shared top-k overlap than the Qwen3-4B student for every teacher: 72.8% versus 87.4% for mathematics, 73.6% versus 87.8% for code, and 67.1% versus 76.2% for instruction following. It also exhibits consistently larger teacher–student discrepancy on the shared support. The instruction-following teacher has the lowest overlap and highest overlap JS divergence in both settings. The shared-support probability masses both remain high, so the mean accessibility difference between the two student scales is modest. Together with the performance ablation in Table 12, these statistics are consistent with accessibility acting as a token-level discount that is most useful in the stronger support-mismatch regime.

Table 13: Per-teacher Novelty statistics averaged over the training trajectory. S-mass and T-mass denote the probability mass assigned by the student and teacher to their shared top-k candidate set.

Student	Teacher	Top-k overlap	S-mass	T-mass	$\widehat{a}$	Overlap JS
Qwen3-1.7B	Math	0.728	0.990	0.992	0.991	0.0386
	Code	0.736	0.991	0.990	0.990	0.0348
	IF	0.671	0.984	0.978	0.981	0.0606
Qwen3-4B	Math	0.874	0.995	0.997	0.996	0.0158
	Code	0.878	0.995	0.996	0.995	0.0131
	IF	0.762	0.991	0.989	0.990	0.0385

F.3 Cross-Domain Routing Example

Aggregate routing mass. For prompt domain d and teacher i, the left panel of Figure 2 reports the response-token routing mass

$$R_{d,i} = \frac{\sum_{x:d(x)=d} \sum_t w_{i,t}}{\sum_{x:d(x)=d} \sum_t \sum_j w_{j,t}}, \tag{12}$$

where the sums include valid response tokens in the labeled-data routing audit. Each row therefore sums to one, and the off-domain mass for domain d is $1 - R_{d,d}$.

Prompt and semantic stages. We examine a Qwen3-1.7B student rollout that combines math reasoning, code verification, and output-format constraints. The complete prompt is:

```
What is the sum of the integers from 1 to 100?  Solve it,
verify your result using Python, and return the final answer
in a strict JSON schema.
```

For exposition, we manually divide the generated response into three semantic stages: mathematical reasoning, Python verification, and JSON formatting. These annotations are applied only after generation and do not enter the routing computation. The main-text trajectory in Figure 2 retains the original coefficient scale and smooths neighboring word positions only for readability.

Per-teacher routing centers. Figure 3 presents the same response at the word-span level. Color intensity is normalized independently for each teacher, interpreted as relative concentration within a teacher across different words. As the figure illustrates, the math teacher concentrates most strongly around the numerical derivation, the code teacher around the Python calculation, and the instruction-following teacher around structural markers and the final JSON-formatted output. This view complements the non-normalized trajectory by making each teacher's local routing center explicit.

F.4 Sensitivity to the Student Top-k Support Size

To assess the robustness of ExpertAlign to its support size, we vary the student's top-k support over $k \in \{1, 4, 8, 16, 32\}$ on the domain-labeled training set with the Qwen3-4B student. We select k on this setting by Overall among these values; $k = 16$ attains the highest score, and since Overall for $k = 1$ to $k = 16$ lies within the error bars, the choice is insensitive to the plateau. Figure 4 shows relatively stable Overall performance from $k = 1$ to $k = 16$, followed by a clear drop at $k = 32$.

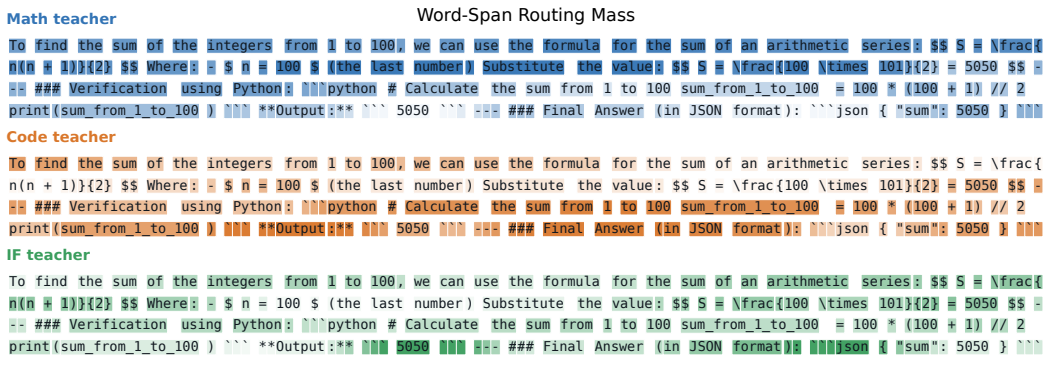


Figure 3: Per-teacher word-span routing patterns for the representative rollout. Color intensity is normalized independently within each teacher; darker spans indicate greater relative routing mass for that teacher. Absolute coefficient magnitudes are shown in Figure 2.

K=1. Mathematics is the only domain that peaks at $k = 1$. In this case, S_t^S contains only the student's most probable token $v_t^\star$, and the expertise and teaching vectors reduce to the scalars, denoted as $e_{i,t} = \log p_t^i(v_t^\star) - \log p_t^{\text{base}}(v_t^\star)$ and $d_{i,t} = \log p_t^i(v_t^\star) - \log p_t^S(v_t^\star)$. The alignment then becomes their product $e_{i,t}d_{i,t}$[1]: a teacher is retained only when its post-training shift and its correction to the student on $v_t^\star$ share the same sign, and retained teachers are weighted by how strongly both quantities move the probability of this single token. This single-token alignment suits mathematics: as shown in Table 10, the mathematics teacher has the lowest predictive entropy and a most sharpened token distribution, so its supervision concentrates on the dominant candidate. Code and instruction following involve more positions with several plausible continuations, where alignment on a single token cannot capture how the teacher redistributes probability among competing candidates.

K=32. Enlarging the support to $k = 32$ degrades all three domains. The added candidates carry little probability mass, as the shared top-16 candidates already cover about 99% of the student's mass according to Table 13. However, each candidate contributes one coordinate to $\mathbf{e}_{i,t}$ and $\mathbf{d}_{i,t}$ regardless of its probability, and log-ratios amplify differences between small probabilities. For example, a tail token whose probability drops from 10^{-3} under the student to 10^{-6} under a sharpened teacher yields $|d_{i,t}(v)| \approx 6.9$, whereas a head token moving from 0.5 to 0.4 yields only 0.22. Because these probabilities are tiny, even small absolute fluctuations also cause large changes in their log-ratios. As more tail candidates enter the support, the alignment $\langle \mathbf{e}_{i,t}, \mathbf{d}_{i,t} \rangle$ is increasingly dominated by near-zero-probability tokens rather than the dominant candidates the student is likely to generate, making both teacher selection and weighting noisy.

F.5 ExpertAlign Selection Coverage and Training Dynamics

Selective routing with broad supervision coverage. Let $K_t = |\mathcal{E}_t|$ denote the number of teachers retained for a valid response token. Figure 5(a) reports the trajectory mean of the token fraction for each value of K_t. Across the four settings, 37.02–51.24% of tokens retain all three teachers; equivalently, ExpertAlign selects a strict subset on 48.76–62.98% of tokens. Most of these selectively routed tokens remain supervised: 47.61–60.50% of all tokens retain one or two teachers, whereas only 0.52–2.48% have $K_t = 0$. Consequently, 97.52–99.48% of valid response tokens receive a signal from at least one teacher. The dominant behavior of the gate is therefore to alter teacher composition, rather than to remove the token from the OPD objective.

Fully filtered tokens increase mildly over training. Although $K_t = 0$ remains rare throughout training, Figure 5(b) shows a positive ordinary-least-squares slope in each setting. The estimated increases per 100 steps are 0.161 and 0.071 percentage points for the unlabeled 1.7B and 4B students,

[1]For $k = 1$, the cosine in Equation 10 between two nonzero scalars is ± 1 and would discard alignment magnitude. Our implementation therefore replaces $c_{i,t}$ with the unnormalized product $\max(e_{i,t}d_{i,t}, 0)$ in Equation 11 for this case; the retained set $\mathcal{E}_t$ is unchanged, and all $k > 1$ use the cosine form.

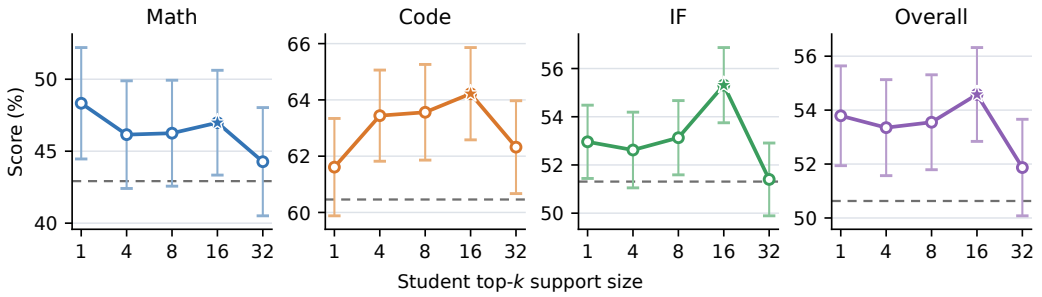
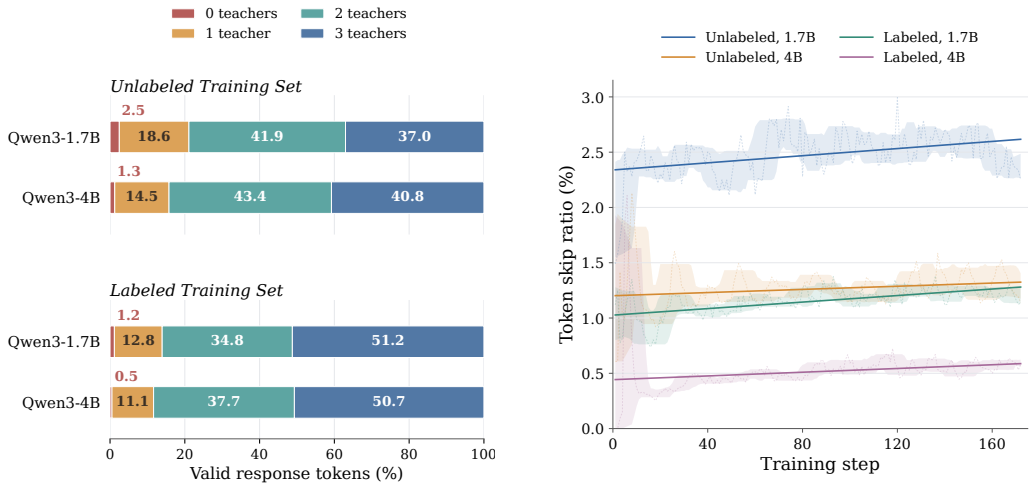


Figure 4: Sensitivity of ExpertAlign to the student's top-k support size. Each panel reports the corresponding domain-average or Overall score; markers denote means and error bars show the reported $\pm$ values. Gray dashed lines denote the performance of Standard MOPD.

and 0.148 and 0.084 percentage points for their labeled-data counterparts. These slopes summarize a mild increasing tendency, and we treat this increase as a descriptive training dynamic and examine whether it is compatible with the geometry of the ExpertAlign gate below.



(a) Retained-teacher count distribution **(b)** All-teacher skip ratio over training

Figure 5: ExpertAlign teacher selection and supervision coverage. (a) Trajectory-mean fraction of valid response tokens with $K_t \in \{0, 1, 2, 3\}$. (b) Evolution of $K_t = 0$ over training. Pale dashed curves are logged observations and lines are ordinary-least-squares fits over the displayed range.

Geometric interpretation. The observed tendency is compatible with a geometric interpretation of the ExpertAlign gate. Define the student's base-relative displacement on the same support as

$$\mathbf{q}_t = \left[\log p_t^S(v) - \log p_t^{\text{base}}(v)\right]_{v \in S_t^S}. \tag{13}$$

For a nonzero specialization vector $\mathbf{e}_{i,t} \neq \mathbf{0}$, the student's normalized progress along that direction as $r_{i,t} = \langle \mathbf{e}_{i,t}, \mathbf{q}_t \rangle / \|\mathbf{e}_{i,t}\|_2^2$. Because $\mathbf{d}_{i,t} = \mathbf{e}_{i,t} - \mathbf{q}_t$, the alignment tested by the gate becomes

$$\langle \mathbf{e}_{i,t}, \mathbf{d}_{i,t} \rangle = \|\mathbf{e}_{i,t}\|_2^2 - \langle \mathbf{e}_{i,t}, \mathbf{q}_t \rangle = \|\mathbf{e}_{i,t}\|_2^2 (1 - r_{i,t}). \tag{14}$$

As the training progresses, the student approaches teacher i and the teaching vector $\mathbf{d}_{i,t}$ contracts, making its inner product with $\mathbf{e}_{i,t}$ more likely to fall below the numerical margin. Besides, when $r_{i,t} \geq 1$, the student's projection reaches or exceeds the teacher's base-relative displacement along this direction, and the alignment also becomes non-positive. Both cases indicate that the student has absorbed the teacher's specialization sufficiently for that teacher to provide little or no positively aligned marginal correction at the current token.

G TRAINING COST ANALYSIS

Beyond the three domain teachers, ExpertAlign also evaluates their shared pre-RL base model and computes top-k distribution statistics for routing, whereas Entropy and Novelty route over the same three-teacher pool as Standard MOPD. To quantify the resulting end-to-end overhead, we compare all three routing metrics against Standard MOPD in the same-size, domain-labeled setting. All measurements use an identical environment with eight NVIDIA H20 GPUs. Because routing methods yield different response lengths, Table 14 reports both measured costs under this common step budget and token-normalized end-to-end costs.

Table 14: End-to-end training cost in the same-size, domain-labeled setting.

Method	Avg. resp. length	GPU-hours	E2E ms/token	Rel. E2E cost
Standard MOPD	4,893	692.9	0.32184	$1.00\times$
MOPD-Router				
w/ Entropy	3,278	524.2	0.36344	$1.13\times$
w/ Novelty	3,496	539.6	0.35079	$1.09\times$
w/ ExpertAlign	4,278	716.9	0.38086	$1.18\times$

Under the common step budget, shorter responses reduce Entropy and Novelty to 524.2 and 539.6 GPU-hours versus Standard MOPD's 692.9. Their token-normalized costs remain close at $1.13\times$ and $1.09\times$, indicating modest routing overhead. ExpertAlign requires 716.9 GPU-hours, a 3.5% measured increase, and $1.18\times$ token-normalized cost. Thus, its shared-base forward pass and top-k routing computation add only modest end-to-end overhead. Measurements exclude initialization, model downloads, validation, and checkpoint I/O outside the per-step timer.