SMOPD: Multi-Reward Reinforcement Learning via Specialize-and-Merge Online Policy Distillation

Wen Wang[‡1,2], **Jiahua Bao**[‡1], **Tu Yongsiqi**[‡1], **Yihao Liu**[‡1], **Haotian Zhou**[‡1], **Haoxuan Ma**[‡1], **Mengyu Zhou**[†1], **Wenkui Fan**[‡1], **Junwei He**[2], **Xiaoxi Jiang**[1] **and Guanjun Jiang**[1]

[1]Qwen Large Model Application Team, Alibaba, [2]University of Chinese Academy of Sciences

[‡]Work done during an internship at Alibaba. [†]Corresponding author.

We aim to improve model performance in multi-reward reinforcement learning training process. Existing Group reward-Decoupled Normalization Policy Optimization (GDPO) has mitigated the issue of reward signals masking one another during direct scalarization by normalizing each reward dimension separately before aggregation. However, our experiments show that GDPO still struggles to balance reward signals with different granularities. Specifically, in some particular training tasks, the model may receive a dense reward that assigns fine-grained scores ranging from 0.1 to 1.0, together with a sparse reward that provides only binary feedback of either 0 or 1. In such cases, we find that the sparse reward may provide an insufficient optimization signal, preventing its corresponding capability from being effectively reinforced. Therefore, ***how can we strengthen the optimization signal from the sparse reward without sacrificing the capability already learned from the fine-grained reward?*** To overcome this limitation, we propose **Specialize-and-Merge Online Policy Distillation (SMOPD)**, a two-stage training method for multi-reward optimization. ***Stage1-Specialize***: SMOPD first employs reward-priority configurations to train multiple reward-specialized teachers, allowing each reward to be learned under conditions where its signal can effectively drive optimization. ***Stage2-Merge***: SMOPD then utilizes online policy distillation to combine the reward-specialized capabilities of these teachers into a single student policy, while maintaining balanced task-level optimization. To validate our method, we conduct experiments on two multi-reward settings: complementary rewards (tool-calling accuracy and format) and conflicting rewards (helpful and harmless rewards). Based on above settings, SMOPD outperforms GDPO across 1.5B, 3B and 7B backbones.

1. Introduction

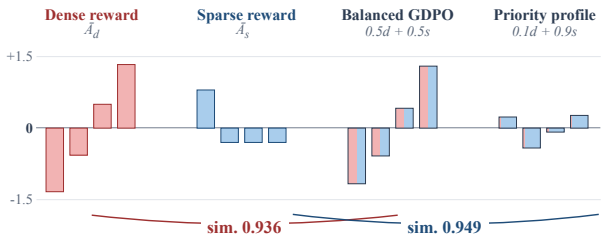
Reinforcement learning from human feedback (RLHF) has become the dominant paradigm for aligning large language models with human preferences Ouyang et al. (2022); Christiano et al. (2017). In multi-reward alignment, prior work commonly adopts GRPO Shao et al. (2024) to optimize a single policy with several reward signals. However, because GRPO first sums different rewards and then computes group-relative advantages, reward dimensions with different scales or reward combinations can mask one another during scalarization, causing the final advantage to lose reward-specific information. GDPO Liu et al. (2026b) mitigates this aggregation-level issue by normalizing each reward dimension separately before aggregation, thereby preserving the contribution of each reward in the training signal. Building on this reward-level decomposition, GD^{2}PO Liu et al. (2026a) further addresses conflicts among reward dimensions by filtering rollouts with severe reward-wise disagreement and reweighting queries according to reward consensus. Despite these improvements, GDPO and GD^{2}PO still focus on how observed reward signals are normalized and combined within a single policy.

This leaves open a different problem: **Reward dimensions can differ not only in scale, but also in how often they provide informative learning signals.** Group-based advantage estimation relies on reward differences among rollouts sampled for the same prompt. Dense reward distributions can rank sampled responses

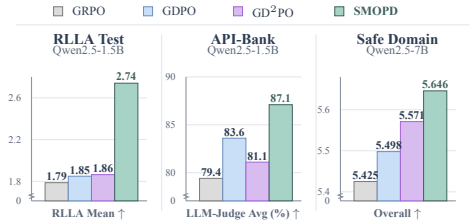
Corresponding author(s): zhoumengyu.zmy@alibaba-inc.com

in most rollout groups, continuously providing reliable optimization signals. Sparse reward distributions, by contrast, may assign the same value to most responses in a group, leaving that reward dimension with little useful within-group variation. In such groups, per-reward normalization cannot create an informative signal where no response-level distinction exists. As a result, under balanced priorities, training is still dominated by dense reward dimensions, while the occasional signal from sparse reward distributions can be overwhelmed before it accumulates. Figure 1(a) illustrates that the balanced GDPO advantage remains closely aligned with the dense reward.

Therefore, single-policy multi-reward RL faces an inherent tension: Raising the priority of a sparse reward to make it learnable inevitably skews the final balance, yet keeping a balanced priority leaves it overshadowed. Our experiments confirm this issue in tool calling Schick et al. (2023); Qin et al. (2024), where a dense accuracy reward is paired with a sparse binary format reward. Under balanced priorities, GDPO behaves similarly to GRPO, with format compliance remaining below 9%. When the format reward is made dominant, the model reliably learns the required output structure, but the resulting policy is optimized under a deliberately skewed objective. This suggests that GDPO alleviates aggregation-level reward masking, yet still struggles with the signal-density imbalance between sparse and dense reward distributions.



(a) Reward-density effect.



(b) Performance comparison.

Figure 1 | **(a)** In a batch of 8 prompt groups with 4 rollouts each, the dense reward vector is $[0, 1, 2, 3]$ in all 8 groups, whereas the sparse reward vector is $[0, 0, 0, 0]$ in 7 groups and $[1, 0, 0, 0]$ in the remaining group. The bars show batch-averaged advantage profiles under balanced weights $(0.5, 0.5)$ and sparse-priority weights $(0.1, 0.9)$. **(b)** Main results comparing SMOPD with multi-reward baselines on RLLA Test (1.5B), API-Bank LLM-Judge (1.5B), and Safe Domain (7B).

To resolve this tension, we propose **Specialize-and-Merge Online Policy Distillation (SMOPD)**, a two-stage method for multi-reward optimization. SMOPD is designed to preserve the strengths learned under dense reward distributions while improving the model's sensitivity to sparse reward distributions. ***Stage1-Specialize***. SMOPD uses reward-priority configurations to train multiple reward-specialized teachers from the same base policy. Instead of balancing all rewards within a single policy, we assign each teacher a reward-priority profile that amplifies its target reward in the optimization signal. In particular, assigning higher priority to a sparse reward counteracts its weaker and less frequent learning signal, enabling the policy to more effectively capture the optimization direction induced by that reward. Figure 1(a) illustrates: a sparse-reward priority profile recovers the sparse reward direction. This allows sparse or hard-to-learn reward distributions to be acquired in a favorable regime, while capabilities learned from dense reward distributions are preserved by complementary teachers. ***Stage2-Merge***. SMOPD merges these reward-specialized teachers into a single student policy through online policy distillation. The student learns from the teachers' reward-specialized behaviors on its own rollouts, while a balanced GDPO anchor maintains task-level optimization over the original multi-reward objective. In this way, SMOPD first acquires different reward strengths through specialization and then balances them through policy-level merging, producing one unified policy instead of a set of separate specialists. Figure 2 summarizes the full workflow.

Experimentally, we validate SMOPD across model scales across 1.5B, 3B and 7B with different model families(Qwen2.5 Qwen Team (2024) and Llama-3.2 Grattafiori et al. (2024)) under two reward structures, including complementary rewards (tool-calling accuracy and format) and conflicting rewards(helpful and harmless rewards) Bai et al. (2022a,b). Our teacher analysis further surfaces structure that a scalarized objective cannot represent.

Our contributions are:

- **SMOPD resolves the reward-balancing tension in multi-reward RL.** Sparse reward distributions are often learnable only when made dominant, but such skewed priorities sacrifice other objectives in a single policy. SMOPD addresses this tension by specializing teachers under complementary reward-priority

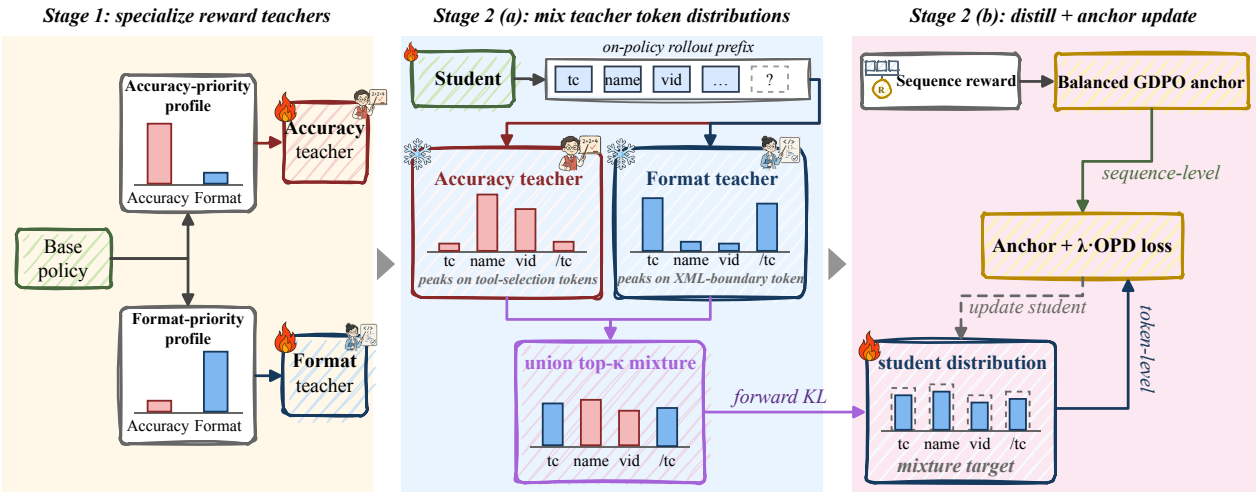


Figure 2 | Overview of SMOPD. *Stage 1*: complementary GDPO priority profiles produce an accuracy teacher and a format teacher from the same base policy. *Stage 2 (a)*: on the student's rollout prefix, frozen teachers emit top-κ next-token distributions. The accuracy teacher peaks on tool-selection tokens, the format teacher peaks on XML boundary tokens, and the union-top-κ mixture retains both. *Stage 2 (b)*: the student is updated through two-level signals. Token-level OPD fits its distribution to the teacher mixture via forward KL, while the sequence-level task anchor optimizes the balanced multi-reward objective.

profiles, then merging them through token-level on-policy distillation with a parameter-free uniform teacher mixture and a sequence-level anchor.

- **SMOPD improves both complementary and conflicting reward settings.** Across three backbones (1.5B, 3B, and 7B), SMOPD consistently exceeds GDPO across different model families. Specifically, in the complementary setting, the improvement is particularly pronounced—peaking at the 1.5B model with a **+48%** composite gain and a dramatic format compliance jump from 8.8% to **97.5%**. In the conflicting setting, SMOPD achieves superior Safe Domain benchmark performance over GDPO across every setting.
- **The merged student surpasses its own teachers.** On safe alignment it exceeds both teachers and the scalarized baseline across three backbones; even at 7B, merging weak teachers yields a clear gain from complementary knowledge unused by scalarized training.

2. Related Work

2.1. Reinforcement Learning for LLMs

RLHF aligns LLMs with human preferences Christiano et al. (2017); Ziegler et al. (2019); Stiennon et al. (2020); Ouyang et al. (2022), classically with PPO Schulman et al. (2017) or its preference-based shortcut DPO Rafailov et al. (2023), though optimizing against learned reward models is prone to over-optimization Gao et al. (2023). GRPO Shao et al. (2024) removes the critic by normalizing rewards within a rollout group, and later variants sharpen this estimator, e.g. DAPO's decoupled clipping and dynamic sampling Yu et al. (2025), GSPO's sequence-level importance ratios Zheng et al. (2025), and simpler REINFORCE-style baselines Ahmadian et al. (2024). For multiple rewards, the standard treatment scalarizes them into one objective and refines the aggregation: GDPO Liu et al. (2026b) normalizes each reward separately and supports priority weights, DVAO Jiang et al. (2026) adapts weights to per-reward gradient magnitudes, and SAW He et al. (2026) reweights objectives by learning speed; other lines constrain or trade off objectives instead of summing them Xu et al. (2024); Dai et al. (2024); Zhou et al. (2024); Wang et al. (2024), or merge separately-trained policies in weight space Wortsman et al. (2022); Ramé et al. (2023, 2024). All of these ultimately ask a single set of parameters to absorb every reward at once, freezing one trade-off at training time; SMOPD instead trains one teacher per reward and defers balancing to a distillation-based merge.

2.2. On-Policy Distillation

Knowledge distillation originally trains a student on a teacher's soft targets Hinton et al. (2015), extended to autoregressive LMs by sequence-level KD on teacher-generated text Kim & Rush (2016). Because teacher-generated data mismatches what the student sees at inference, GKD Agarwal et al. (2024) distills on the student's *own* rollouts—on-policy distillation—with the divergence choice studied by MiniLLM Gu et al. (2024), DistiLLM Ko et al. (2024), and f-divergence KD Wen et al. (2023), and rollout selection refined by PG-OPD Zhao et al. (2026); we follow this line and use forward KL on student rollouts. Recent work scales OPD to *multiple teachers*, fusing separately-trained domain-specialized policies (math, coding, instruction following) into one model more effectively than reward mixing, cascade RL, or parameter merging Ma et al. (2026); MiMo LLM-Core Xiaomi (2026) scales this recipe to frontier post-training, and G-OPD Yang et al. (2026) merges domain-specialized policies back into a shared base. Because each prompt belongs to a single domain, all of these route it to the one teacher that owns it. Our problem is orthogonal: *within a single domain*, several rewards act on the *same* prompt at once, so no routing can separate them; SMOPD instead combines reward-specialized teachers at every token, turning on-policy distillation into a mechanism for multi-reward balancing.

3. Method: SMOPD Framework

The core philosophy of SMOPD is to let each reward be learned where it is easiest, in a *teacher* trained to prioritize it, and then to transfer the teachers' competence into a single student at the granularity where rewards actually act: *individual tokens*. This section presents the two stages in turn.

3.1. Stage 1: Reward-Specialized Teacher Training

Group-based methods sample G rollouts $\{o_1, \ldots, o_G\}$ per prompt and score each rollout o_i with K reward dimensions $r_i = (r_i^{(1)}, \ldots, r_i^{(K)})$; we write $\mu^{(k)}, \sigma^{(k)}$ for the group mean and standard deviation of dimension k. GRPO Shao et al. (2024) normalizes the *summed* reward within the group, $\hat{A}_i^{\text{GRPO}} = (\sum_k r_i^{(k)} - \mu_S)/(\sigma_S + \epsilon)$, where μ_S, σ_S are the group statistics of the sum and ϵ is a small stabilizing constant. GDPO Liu et al. (2026b) instead (i) normalizes each dimension within the group, (ii) aggregates the resulting advantages under priority weights $\mathbf{w} = (w_1, \ldots, w_K)$ (equal by default), and (iii) applies a final *batch*-wise whitening over all responses in the update:

$$\hat{A}_i^{(k)} = \frac{r_i^{(k)} - \mu^{(k)}}{\sigma^{(k)}}, \tag{1}$$

$$A_i^{\text{sum}} = \sum_{k=1}^{K} w_k \cdot \hat{A}_i^{(k)}, \tag{2}$$

$$\hat{A}_i^{\text{GDPO}} = \frac{A_i^{\text{sum}} - \mu_{\text{batch}}(A^{\text{sum}})}{\sigma_{\text{batch}}(A^{\text{sum}}) + \epsilon}. \tag{3}$$

Both estimators train the policy π_θ with the standard clipped surrogate objective on these advantages Schulman et al. (2017); Shao et al. (2024).

GDPO's priority weights thus let a single policy be steered toward a selected trade-off. SMOPD repurposes this control as *reward-priority specialization*: a set of complementary priority profiles constructs multiple teachers, one for each reward dimension, before they are merged in Stage 2. For a reward dimension k we wish to specialize on, we set $w_k \gg w_j$ for $j \neq k$:

$$\hat{A}_{\text{teacher},k} = w_k^{\text{high}} \cdot \hat{A}^{(k)} + \sum_{j \neq k} w_j^{\text{low}} \cdot \hat{A}^{(j)} \tag{4}$$

For example, with $K = 2$ (accuracy and format), we train a **format teacher** with $\mathbf{w} = (0.1, 0.9)$, amplifying the format reward signal by 9× relative to accuracy, and an **accuracy teacher** with $\mathbf{w} = (0.9, 0.1)$, and vice versa.

Because per-reward normalization gives each $\hat{A}^{(k)}$ zero mean and unit variance, the 9× weight ratio directly translates to a 9× expected gradient contribution from the favored reward (the normalized advantages are

placed on a common scale, so the weights alone set their relative influence). This makes the model "zoom in" on its target dimension, reliably maximizing it; the non-target dimensions may degrade (as for the accuracy teacher) or, when the dimensions are not in conflict, be retained (as for the format teacher, Section 4.2). Either way the teachers become *complementary*, each contributing a distinct strength to the subsequent merge.

3.2. Stage 2: Multi-Teacher Online Policy Distillation

Given M teacher teachers $\{\pi_1, \ldots, \pi_M\}$, SMOPD distills their token-level knowledge into a student π_θ. Unlike offline distillation, the student generates its own responses (on-policy), and teachers provide top-κ log-probabilities on the student's sequences—following the on-policy distillation paradigm Agarwal et al. (2024), which avoids the exposure bias of distilling on a fixed teacher-generated corpus.

Teacher mixture distribution. At each token position t in a student-generated response, the teacher target is the α-weighted mixture of the M teachers' next-token distributions:

$$p_T^{\text{mix}}(v \mid s_t) \propto \sum_{m=1}^{M} \alpha_m \, p_{\pi_m}(v \mid s_t), \tag{5}$$

where v ranges over vocabulary tokens, s_t is the student's generated prefix up to position t, and α_m are the mixture weights. SMOPD uses **uniform** weights $\alpha_m = 1/M$ (e.g. $[0.5, 0.5]$ for two teachers), the simplest and, as we show, strongest choice in our setting; input-dependent gating alternatives are explored in Supplementary material E.

Operating over the full vocabulary for every teacher at every position is prohibitive, so we realize Eq. equation 5 with a *component-wise top-κ* construction ($\kappa = 16$ by default, distinct from the reward-dimension count K; sensitivity is studied in Section 4.3). Each teacher m emits only its own top-κ token set $\mathcal{T}_m^\kappa$ with log-probabilities $\log p_{\pi_m}$. We shift each by $\log \alpha_m$, pool the $M \times \kappa$ candidates, and keep the κ largest as the mixture target:

$$\tilde{p}_T^{\text{mix}}(\cdot \mid s_t) = \underset{\substack{m=1,\ldots,M \\ v \in \mathcal{T}_m^\kappa}}{\text{top-}\kappa} \left[\log \alpha_m + \log p_{\pi_m}(v \mid s_t) \right]. \tag{6}$$

This is a fast approximation to the exact union-and-renormalize mixture ($\log \sum_m \alpha_m p_m$): a token in several teachers' top-κ sets may appear more than once among the candidates, and tokens outside every teacher's top-κ set are dropped. With high-quality teachers whose top-κ mass dominates, the two coincide closely. The result is a single per-position top-κ target that jointly carries both teachers.

Distillation loss. The student minimizes the forward KL divergence from the (top-κ) teacher mixture, evaluated over the mixture's support:

$$\mathcal{L}_{\text{OPD}} = \sum_t \text{KL}\left(\tilde{p}_T^{\text{mix}}(\cdot \mid s_t) \,\middle\|\, p_\theta(\cdot \mid s_t) \right) \tag{7}$$

We use forward (rather than reverse) KL deliberately: forward KL is mode-covering Gu et al. (2024); Ko et al. (2024), encouraging the student to place mass on *all* high-probability tokens of the teacher mixture—important here because different teachers may be confident about different tokens (e.g., format wrappers vs. function arguments) at the same position, and the union-top-κ target in Eq. equation 6 preserves both. Section 4.3 ablates both choices, while Supplementary material C details the sampled-token reverse-KL estimator.

GDPO anchor. To prevent the student from only imitating teacher distributions without task-level grounding, we add a GDPO loss with equal weights $\mathbf{w}_{\text{equal}} = (1/K, \ldots, 1/K)$:

$$\mathcal{L}_{\text{anchor}} = \mathcal{L}_{\text{GDPO}}(\pi_\theta; \mathbf{w}_{\text{equal}}), \tag{8}$$

where $\mathcal{L}_{\text{GDPO}}(\pi_\theta; \mathbf{w})$ denotes the clipped policy-gradient loss driven by the GDPO advantage of Eq. equation 3 under weights $\mathbf{w}$.

Total objective. The student optimizes:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{anchor}} + \lambda \cdot \mathcal{L}_{\text{OPD}} \tag{9}$$

where λ controls the distillation strength (default $\lambda = 1.0$). The anchor provides sequence-level direction ("this response should be upweighted/downweighted"), while OPD provides token-level guidance ("at this position, the distribution should look like this mixture of teachers").

		RLLA-4K test (in-domain)			**Generalization** (held-out tool use)				
		Acc	**Format**	**RLLA**	**BFCL**	**API-Bank (LLM-Judge)** (%)			
Backbone	**Method**	**Reward**	**Pass**	**Mean**	**AST**	**L1**	**L2**	**L3**	**Avg**
Qwen2.5-1.5B	GRPO Baseline	1.737	5.0%	1.787	68.6%	79.2	71.6	84.0	79.4
	GDPO Baseline	1.761	8.8%	1.849	72.6%	84.7	74.6	**84.7**	83.6
	GD^{2}PO Baseline	**1.776**	8.8%	1.864	**73.3%**	82.2	71.6	82.4	81.1
	Accuracy Teacher [0.9,0.1]	1.779	1.2%	1.791	70.5%	81.2	73.1	84.0	80.9
	Format Teacher [0.1,0.9]	1.771	97.5%	2.746	70.5%	85.5	73.1	67.9	80.2
	SMOPD	1.765	**97.5%**	**2.740**	70.7%	**90.0**	**82.1**	80.9	**87.1**
Qwen2.5-3B	GRPO Baseline	**1.783**	**97.5%**	**2.758**	75.8%	81.5	65.7	47.3	72.2
	GDPO Baseline	1.753	**97.5%**	2.728	75.3%	80.7	**74.6**	64.1	76.4
	GD^{2}PO Baseline	1.732	**97.5%**	2.707	76.5%	82.0	67.2	61.1	75.7
	Accuracy Teacher [0.9,0.1]	1.795	95.0%	2.745	77.2%	83.2	68.7	61.8	76.9
	Format Teacher [0.1,0.9]	1.740	97.2%	2.712	76.1%	84.2	70.2	62.6	77.9
	SMOPD	1.763	**97.5%**	2.738	**77.2%**	**83.5**	64.2	**65.7**	**77.4**

Table 1 | Main results on complementary rewards setting (RLLA), across Qwen2.5-{1.5B, 3B}-Instruct. **In-domain** columns use the held-out RLLA-4K test split; **Generalization** columns are held-out tool-use benchmarks. For API-Bank, `qwen3.7-plus` serves as the semantic judge for original exact-match failures while exact-match successes are retained; All metrics are defined in Section 4.1; **Bold** marks the best and underline the second best among the baselines and SMOPD;

4. Experiments

4.1. Experimental Setup

Training. We train SMOPD on five backbones. Within each setting, the baselines, reward-specialized teachers, and SMOPD students share the same training recipe, differing only in the GDPO priority profile and, for SMOPD, the added distillation loss; all runs utilize the verl framework Sheng et al. (2025) on 8 H100 GPUs. **Complementary rewards(RLLA).** We train Qwen2.5-{1.5B, 3B}-Instruct Qwen Team (2024) with G=8 rollouts per prompt on the RLLA-4K *training split* from ToolRL Qian et al. (2025). The *accuracy reward* ($r_{\text{acc}} \in [-3, 3]$) scores function-name and parameter matching, while the binary *format reward* ($r_{\text{fmt}} \in \{0, 1\}$) checks the required XML structure (`<think>`, `<tool_call>`, and `<response>` tags). The format and accuracy teachers are trained with complementary priority profiles, $\mathbf{w} = (0.1, 0.9)$ and $(0.9, 0.1)$, respectively; we additionally conduct an ablation study on other priority configurations in Supplementary material D. **Conflicting rewards(helpful + harmless).** We train Qwen2.5-{3B, 7B}-Instruct and Llama-3.2-3B-Instruct with G=4 on prompt-only Alpaca Taori et al. (2023). Useful and harmless rewards are produced by dual reward models trained on PKU-SafeRLHF preference data Dai et al. (2024), and the corresponding teachers employee the complementary profiles, $[0.7, 0.3]$ and $[0.3, 0.7]$. Full training hyperparameters are provided in Supplementary material A (Tables 5 and 6).

Evaluation. All models are evaluated with vLLM Kwon et al. (2023) using temperature 0.0 and top-p 1.0; **Complementary rewards(RLLA).** On the held-out RLLA-4K test split (80 prompts), **RLLA Mean** is the primary composite metric $r_{\text{acc}} + r_{\text{fmt}}$; we additionally report its **Acc Reward** component and the binary **Format Pass** rate. Generalization is evaluated on two held-out tool-use benchmarks: **BFCL AST** is BFCL-v4 Patil et al. (2024) function-calling accuracy averaged over the non-live and live AST categories, and **API-Bank (LLM-Judge)** Li et al. (2023) measures functional accuracy on Level-1/2/3. For API-Bank, exact-match successes remain correct and an LLM judge Zheng et al. (2023) reviews only exact-match failures for functionally equivalent tool calls; we report **L1–L3** and micro-accuracy over all 597 items (**Avg**). The BFCL category breakdown is in Supplementary material G, and the complete API-Bank protocol, original strict scores, and failure analysis are in Supplementary material F. **Conflicting rewards(helpful + harmless).** Every model is scored by the same dual reward models on held-out HH-RLHF Bai et al. (2022a), PKU-SafeRLHF, and Alpaca prompts (mean@1).

Backbone	Method	HH-RLHF			PKU-SafeRLHF			Alpaca			Overall
		U.	H.	Avg	U.	H.	Avg	U.	H.	Avg	**Avg**
Qwen2.5-3B	GRPO	4.400	5.690	5.045	5.372	6.878	6.125	5.281	6.020	5.650	5.607
	GDPO	4.382	5.704	5.043	5.373	**6.883**	6.128	5.247	6.001	5.624	5.598
	GD^{2}PO	4.485	5.730	5.107	5.406	6.879	6.143	5.379	6.023	5.701	5.650
	Useful Teacher	4.448	5.472	4.960	5.303	6.700	6.001	5.407	5.736	5.572	5.511
	Harmless Teacher	4.320	5.784	5.052	5.375	6.919	6.147	5.149	6.176	5.662	5.620
	SMOPD	**4.511**	**5.743**	**5.127**	**5.426**	6.869	**6.147**	**5.392**	**6.071**	**5.732**	**5.669**
Qwen2.5-7B	GRPO	4.237	5.469	4.853	5.386	6.834	6.110	4.976	5.647	5.312	5.425
	GDPO	4.318	5.439	4.878	5.329	6.766	6.048	5.323	5.813	5.568	5.498
	GD^{2}PO	4.379	5.511	4.945	5.390	6.801	6.096	**5.434**	5.909	5.672	5.571
	Useful Teacher	4.515	5.313	4.914	5.430	6.658	6.044	5.513	5.572	5.543	5.500
	Harmless Teacher	4.220	5.607	4.913	5.354	6.889	6.122	5.021	6.041	5.531	5.522
	SMOPD	**4.475**	**5.656**	**5.065**	**5.448**	**6.837**	**6.143**	5.421	**6.036**	**5.729**	**5.646**
Llama-3.2-3B	GRPO	4.264	5.510	4.887	5.117	6.645	5.881	5.340	5.891	5.615	5.461
	GDPO	4.414	5.642	5.028	5.315	6.761	6.038	5.366	5.998	5.682	5.583
	GD^{2}PO	**4.438**	**5.657**	**5.047**	5.319	**6.776**	6.047	**5.429**	**6.012**	**5.721**	**5.605**
	Useful Teacher	4.174	5.177	4.676	5.198	6.512	5.855	5.296	5.489	5.393	5.308
	Harmless Teacher	3.917	5.271	4.594	5.046	6.626	5.836	4.869	5.724	5.296	5.242
	SMOPD	4.437	5.633	5.035	**5.349**	6.756	**6.052**	5.406	5.962	5.684	5.590

Table 2 | Main results on conflicting rewards(Safe-alignment benchmarks) across Qwen2.5-3B, 7B-Instruct, and Llama-3.2-3B-Instruct, reported per benchmark with separate Useful (U.) and Harmless (H.) scores and their average (Avg); **Overall** is the mean Avg across the three benchmarks.

We report per-benchmark **Useful**, **Harmless**, and their average; **Overall** is the mean of these averages across the three benchmarks.

4.2. Main Results

Reward-priority specialization makes sparse reward distributions learnable. As shown in the 1.5B setting of Table 1 (rows 2–4), GDPO struggles to optimize the format reward under balanced priorities, achieving only 8.8% format compliance on the RLLA test set. This is because the binary format reward follows a sparse reward distribution: in most rollout groups, responses receive identical format scores, leaving little within-group variation for group-based RL to exploit. Although the non-zero format score indicates that the model can occasionally generate correctly formatted responses, this sparse signal is overwhelmed by the denser accuracy reward distribution during balanced multi-reward optimization. By increasing the priority of the format reward, the format-specialized teacher ($\mathbf{w} = [0.1, 0.9]$) amplifies this otherwise underutilized signal and raises format compliance to 97.5%. Importantly, this gain does not come at the cost of accuracy performance. It shows that reward-priority specialization can make the sparse reward distribution learnable while preserving the dense-reward capability.

SMOPD under complementary rewards. We first study the complementary reward setting, where accuracy and format rewards supervise different aspects of tool-calling behavior. Table 1 reports results on both backbones. On Qwen2.5-1.5B, SMOPD achieves an RLLA score of 2.740, yielding a **+48%** improvement over the GDPO baseline (1.849), while preserving the structured-output capability learned by the format-specialized teacher (97.5% format compliance). Meanwhile, the student retains accuracy-oriented tool-use ability: BFCL AST reaches 70.7%, and API-Bank LLM-judge Avg reaches 87.1%—the best among all 1.5B methods in Table 1 and **+3.5%** points above the strongest scalarized baseline. We report the API-Bank LLM-judge metric because the original strict exact-match protocol can penalize functionally equivalent tool calls due to superficial differences in argument formatting or values; therefore, we apply semantic judging only to exact-match failures while preserving exact successes. The complete judging procedure and the corresponding strict exact-match results are provided in Supplementary material F. These results show that SMOPD can merge complementary teacher

capabilities instead of collapsing them into a single averaged behavior.

SMOPD under conflicting rewards. We further evaluate SMOPD under conflicting rewards, where helpfulness and harmlessness require balancing competing alignment objectives (Section 4.2). Across three backbones (Table 2), SMOPD consistently improves over scalarized baselines by merging reward-specialized capabilities; adaptive teacher mixing variants are compared in Supplementary material E. On Qwen2.5-3B, SMOPD achieves the best Overall score (**5.669**), outperforming the strongest scalarized baseline GD^{2}PO (5.650) and both single-reward teachers. The gain becomes more evident on Qwen2.5-7B: although each individual teacher provides only marginal improvement over GDPO (5.500 and 5.522 versus 5.498), SMOPD further improves the merged policy to 5.646, exceeding all baselines and teachers. This suggests that even under conflicting objectives, reward-specialized teachers retain complementary capabilities that can be recovered through policy-level merging. On Llama-3.2-3B, where teacher specialization is the most challenging, both individual teachers degrade substantially, yet SMOPD recovers the performance to 5.590, approaching the strongest scalarized baseline GD^{2}PO (5.605) while surpassing GDPO (5.583). The remaining gap to GD^{2}PO is mainly attributed to the task anchor rather than the teacher-merging mechanism: SMOPD uses GDPO as its sequence-level anchor, whereas GD^{2}PO improves the scalarized optimization objective itself through a refined advantage estimator. Therefore, improving the anchor provides an orthogonal direction to further enhance SMOPD, while the consistent gain over GDPO demonstrates the effectiveness of reward-specialized capability merging.

4.3. Ablation and Analysis

Ablating the Task Anchor. SMOPD combines two training signals operating at different levels of granularity. The *sequence-level* GDPO anchor determines whether a sampled response should be upweighted or down-weighted according to reward-normalized advantages, providing task-level optimization direction but only a coarse scalar signal for the entire response. In contrast, *token-level* OPD distillation shapes the student's distribution toward the teacher mixture at every generation step, providing dense supervision through hundreds of token-level gradient signals within a single response.

This combination is particularly valuable for format compliance, where the reward is binary (0/1 per response) but the required behavior—correct XML structure spanning many tokens—needs position-level guidance. The GDPO anchor alone cannot teach format because it reduces to a single advantage shared by all tokens in a response. OPD provides the missing token-level granularity: the format teacher's high confidence on XML wrapper tokens (`<tool_call>`, `</tool_call>`) and the accuracy teacher's specialization on function-argument tokens jointly shape the student's behavior at each position.

To test whether both signals are necessary, we ablate the anchor on Qwen2.5-7B safe alignment at the controlled setting $\kappa = 32$, optimizing only the OPD objective, $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{OPD}}$ (Table 3). OPD alone already improves over the individual teachers, reaching 5.544 Overall compared with 5.500 for the useful teacher and 5.522 for the harmless teacher. However, it remains close to teacher performance, suggesting that pure distillation is limited by the quality of the teacher mixture it imitates. Reintroducing the anchor lifts SMOPD to 5.639 Overall, exceeding both teachers. The anchor therefore supplies an additional task-level optimization signal beyond teacher imitation, allowing the student to improve after absorbing complementary teacher behaviors. Supplementary material B formalizes this distinction: the forward-KL objective in Eq. equation 7 reaches its minimum at the teacher mixture, where the distillation gradient vanishes, whereas the policy-gradient anchor can continue optimizing task reward.

Method	Useful	Harmless	Overall
GDPO baseline (anchor only)	4.990	6.006	5.498
Useful teacher	5.153	5.848	5.500
Harmless teacher	4.865	6.179	5.522
OPD only (no anchor)	<u>5.011</u>	<u>6.077</u>	<u>5.544</u>
SMOPD (OPD + anchor)	**5.099**	**6.178**	**5.639**

Table 3 | Ablating the task anchor on Qwen2.5-7B-Instruct safe alignment at the controlled setting $\kappa = 32$ (dual-RM mean@1; **Overall** = mean of Useful and Harmless).

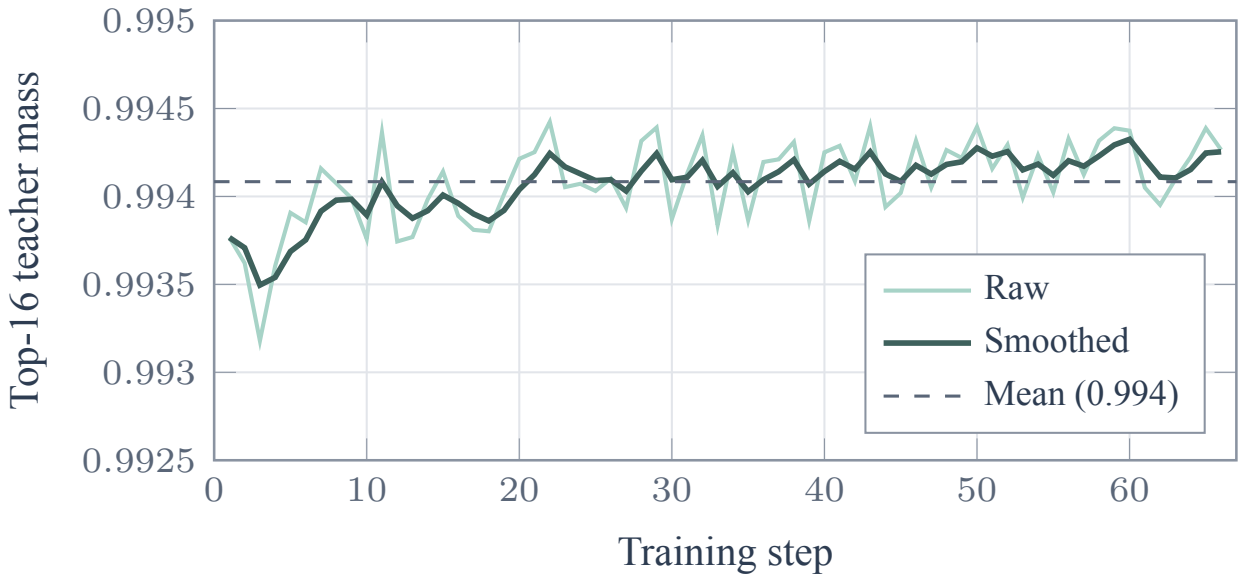


Figure 3 | Top-16 teacher probability mass during Qwen2.5-7B safe-alignment training; light and dark curves show the raw and smoothed values.

Ablating OPD Design Choices. The top-κ approximation controls the number of teacher candidates retained at each position, trading a broader approximation to the teacher distributions for distillation cost. We vary $\kappa \in \{16, 32, 64\}$ for SMOPD on Qwen2.5-7B-Instruct safe alignment while keeping the training and evaluation protocol unchanged, and separately compare the sampled-token reverse-KL update detailed in Supplementary material C (Table 4). Forward-KL performance remains highly stable across support sizes, with Overall varying by only 0.007 between $\kappa = 16$ and $\kappa = 64$. To understand why a small support is sufficient, we measure the teacher probability mass retained by the top-κ approximation. The top-16 support already preserves approximately 0.994 of the teacher mass throughout training (Figure 3), explaining why increasing κ provides little additional benefit.

The sampled-token reverse-KL variant reaches 5.563 Overall, 0.082 below forward KL with $\kappa = 16$. This gap is consistent with the different behaviors of the two divergence directions in our multi-teacher setting. Forward KL is mode-covering, encouraging the student to preserve the diverse high-probability regions in the teacher mixture, which is important when different teachers contribute distinct behaviors. In contrast, reverse KL is mode-seeking and may concentrate on a subset of dominant teacher modes, potentially losing specialized capabilities from other teachers. Thus, forward KL is preferable in this setting, although this single-backbone result does not imply that reverse KL is universally inferior.

Objective	**Support**	**Useful**	**Harmless**	**Overall**
GDPO baseline	–	4.990	6.006	5.498
Forward KL	$\kappa = 16$	**5.115**	6.176	**5.646**
Forward KL	$\kappa = 32$	5.099	<u>6.178</u>	5.639
Forward KL	$\kappa = 64$	<u>5.102</u>	**6.181**	<u>5.642</u>
Reverse KL (PG)	sampled	5.034	6.093	5.563

Table 4 | Top-κ support and KL-direction ablation on Qwen2.5-7B safe alignment. Useful and Harmless are averaged across HH-RLHF, PKU-SafeRLHF, and Alpaca; Overall is their mean.

5. Conclusion

We identified a reward-density imbalance in multi-reward reinforcement learning: even after per-reward normalization, sparse rewards may provide too little within-group variation to effectively influence a shared policy. To overcome this limitation, we introduced SMOPD, which decouples capability acquisition from reward balancing by training reward-specialized teachers and merging their capabilities into a single student through online policy distillation with a balanced task anchor. Across complementary rewards and conflicting rewards settings, SMOPD consistently improves over GDPO on 1.5B, 3B, and 7B backbones. These results highlight the effectiveness of our method in multi-reward optimization: reward-specific specialization preserves complementary capabilities, while subsequent merging produces a balanced final policy without over-prioritizing any single objective.

References

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

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting REINFORCE-style optimization for learning from human feedback in LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, and Jared Kaplan. Training a helpful and harmless assistant with reinforcement learning from human feedback, 2022a.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional AI: Harmlessness from AI feedback, 2022b.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Juntao Dai, Xuehai Pan, Ruiyang Sun, Jiaming Ji, Xinbo Xu, Mickel Liu, Yizhou Wang, and Yaodong Yang. Safe RLHF: Safe reinforcement learning from human feedback. In *International Conference on Learning Representations (ICLR)*, 2024.

Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, pp. 10835–10866, 2023.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, et al. The llama 3 herd of models, 2024.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. MiniLLM: Knowledge distillation of large language models. In *International Conference on Learning Representations (ICLR)*, 2024.

Yuchen He, Baolong Bi, Shenghua Liu, Huaming Liao, Yuyao Ge, Bolin Wan, Siqian Tong, Juan Chen, Jiafeng Guo, and Xueqi Cheng. SAW: Stage-aware dynamic weighting for multi-objective reinforcement learning in large language models, 2026.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*, 2015.

Guochao Jiang, Jingyi Song, Guofeng Quan, Chuzhan Hao, Guohua Liu, and Yuewei Zhang. DVAO: Dynamic variance-adaptive advantage optimization for multi-reward reinforcement learning, 2026.

Yoon Kim and Alexander M. Rush. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1317–1327, 2016.

Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. DistiLLM: Towards streamlined distillation for large language models. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pp. 611–626, 2023.

Minghao Li, Yingxiu Zhao, Bowen Yu, Feifan Song, Hangyu Li, Haiyang Yu, Zhoujun Li, Fei Huang, and Yongbin Li. API-bank: A comprehensive benchmark for tool-augmented LLMs. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023.

Haotian Liu, Yihao Liu, Jingwei Ni, Siyuan Huang, Xinpeng Liu, Pengyu Cheng, Jiajun Song, Ruijin Ding, Junfeng Li, Zhechao Yu, Mengyu Zhou, Hongteng Xu, Xiaoxi Jiang, and Guanjun Jiang. GD^{2}PO: Mitigating multi-reward conflicts via group-dynamic reward-decoupled policy optimization, 2026a.

Shih-Yang Liu, Xin Dong, Ximing Lu, Shizhe Diao, Peter Belcak, Mingjie Liu, Min-Hung Chen, Hongxu Yin, Yu-Chiang Frank Wang, Kwang-Ting Cheng, Yejin Choi, Jan Kautz, and Pavlo Molchanov. GDPO: Group reward-decoupled normalization policy optimization for multi-reward RL optimization, 2026b.

LLM-Core Xiaomi. MiMo-V2-Flash technical report, 2026.

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

Wenhan Ma, Jianyu Wei, Liang Zhao, Hailin Zhang, et al. MOPD: Multi-teacher on-policy distillation for capability integration in LLM post-training, 2026.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pp. 27730–27744, 2022.

Shishir G. Patil, Tianjun Zhang, Xin Wang, and Joseph E. Gonzalez. Gorilla: Large language model connected with massive APIs. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Cheng Qian, Emre Can Acikgoz, Qi He, Hongru Wang, Xiusi Chen, Dilek Hakkani-Tür, Gokhan Tur, and Heng Ji. ToolRL: Reward is all tool learning needs. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, Sihan Zhao, Lauren Hong, Runchu Tian, Ruobing Xie, Jie Zhou, Mark Gerstein, Dahai Li, Zhiyuan Liu, and Maosong Sun. ToolLLM: Facilitating large language models to master 16000+ real-world APIs. In *International Conference on Learning Representations*, 2024.

Qwen Team. Qwen2.5 technical report, 2024.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Alexandre Ramé, Guillaume Couairon, Mustafa Shukor, Corentin Dancette, Jean-Baptiste Gaya, Laure Soulier, and Matthieu Cord. Rewarded soups: towards Pareto-optimal alignment by interpolating weights fine-tuned on diverse rewards. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Alexandre Ramé, Johan Ferret, Nino Vieillard, Robert Dadashi, Léonard Hussenot, Pierre-Louis Cedoz, Pier Giuseppe Sessa, Sertan Girgin, Arthur Douillard, and Olivier Bachem. WARP: On the benefits of weight averaged rewarded policies, 2024.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

John Schulman. Approximating KL divergence. http://joschu.net/blog/kl-approx.html, 2020.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 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, 2024.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. HybridFlow: A flexible and efficient RLHF framework. In *Proceedings of the Twentieth European Conference on Computer Systems (EuroSys)*, 2025.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. In *Advances in Neural Information Processing Systems*, volume 33, pp. 3008–3021, 2020.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following LLaMA model. https://github.com/tatsu-lab/stanford_alpaca, 2023.

Haoxiang Wang, Yong Lin, Wei Xiong, Rui Yang, Shizhe Diao, Shuang Qiu, Han Zhao, and Tong Zhang. Arithmetic control of LLMs for diverse user preferences: Directional preference alignment with multi-objective rewards. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024.

Yuqiao Wen, Zichao Li, Wenyu Du, and Lili Mou. f-divergence minimization for sequence-level knowledge distillation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pp. 10817–10834, 2023.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3):229–256, 1992.

Mitchell Wortsman, Gabriel Ilharco, Samir Yitzhak Gadre, Rebecca Roelofs, Raphael Gontijo Lopes, Ari S. Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, 2022.

Tengyu Xu, Eryk Helenowski, Karthik Abinav Sankararaman, Di Jin, Kaiyan Peng, Eric Han, Shaoliang Nie, Chen Zhu, Hejia Zhang, Wenxuan Zhou, Zhouhao Zeng, Yun He, Karishma Mandyam, Arya Talabzadeh, Madian Khabsa, Gabriel Cohen, Yuandong Tian, Hao Ma, Sinong Wang, and Han Fang. The perfect blend: Redefining RLHF with mixture of judges, 2024.

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

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Juncai Liu, Lingjun Liu, Xin Liu, et al. DAPO: An open-source LLM reinforcement learning system at scale. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Qingfei Zhao, Huan Song, Shuyu Tian, Jiawei Shao, and Xuelong Li. Prefix-guided on-policy distillation: Mining golden trajectories from rollouts, 2026.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, Jingren Zhou, and Junyang Lin. Group sequence policy optimization, 2025.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-Bench and chatbot arena. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Zhanhui Zhou, Jie Liu, Jing Shao, Xiangyu Yue, Chao Yang, Wanli Ouyang, and Yu Qiao. Beyond one-preference-fits-all alignment: Multi-objective direct preference optimization. In *Findings of the Association for Computational Linguistics: ACL 2024*, 2024.

Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences, 2019.

Accuracy + format (RLLA)	**Value**
Base models	Qwen2.5-{1.5B, 3B}-Instruct
Training data	RLLA-4K train split (ToolRL)
Reward dimensions	$r_{\text{acc}} \in [-3, 3]$, $r_{\text{fmt}} \in \{0, 1\}$
GD^{2}PO variant	hard sign-conflict filtering
Teacher weights	$[0.9, 0.1]$ acc / $[0.1, 0.9]$ fmt
Anchor weights	equal, $[0.5, 0.5]$
Teacher mixing (fixed)	$\alpha_m = 1/M = 0.5$
Training steps	150 (1.5B); 100 (3B)
Rollouts per prompt (G)	8
Learning rate	1×10^{-6}
OPD top-κ	16
OPD coefficient (λ)	1.0
Distillation loss	forward KL, top-κ support
Eval temperature / top-p	0.0 / 1.0
Max new tokens	1024
Eval protocol	greedy decoding, single pass
GPUs / tensor parallelism	8 / 1
Hardware	8× NVIDIA H100 GPUs

Table 5 | Training and evaluation configuration for the accuracy + format experiments (both backbones).

Safe alignment (helpful + harmless)	**Value**
Base models	Qwen2.5-{3B, 7B}-Instruct, Llama-3.2-3B-Instruct
Training prompts	Alpaca (prompt-only) train split
Useful reward	Qwen2.5-7B-SafeRLHF-RM
Harmless reward	Qwen2.5-7B-SafeRLHF-CM
GD^{2}PO variant	hard sign-conflict filtering
Teacher weights	$[0.7, 0.3]$ useful / $[0.3, 0.7]$ harmless
Anchor weights	equal, $[0.5, 0.5]$
Teacher mixing (fixed)	$\alpha_m = 1/M = 0.5$
Train batch / mini-batch	512 / 128
Rollouts per prompt	4
Training steps	100 (3B); 66 (7B)
Rollout temperature / top-p	0.7 / 1.0
Learning rate	1×10^{-6}
OPD top-κ, λ	16, 1.0
Eval benchmarks	HH-RLHF (8,520), PKU-SafeRLHF (8,211), Alpaca (512)
Eval protocol	mean@1 per benchmark
Hardware	8× NVIDIA H100 GPUs

Table 6 | Training and evaluation configuration for the safe-alignment experiments (all three backbones).

A. Training and Evaluation Details

All models are trained with the verl framework using Ray-based distributed training. Table 5 lists the configuration of the accuracy+format (RLLA) experiments and Table 6 that of the safe-alignment experiments. Within each setting, GDPO, the teachers, and the SMOPD students share the same recipe, differing only in the GDPO reward weights (and, for SMOPD, the added distillation loss); GD^{2}PO changes only the advantage estimator, as detailed below. The RLLA runs use 150 steps for Qwen2.5-1.5B and 100 for Qwen2.5-3B; safe-alignment runs use 100 steps for the 3B backbones and 66 steps for Qwen2.5-7B. The confidence-failure gates (Sec. E) use weight floor $\phi = 0.2$, exponents $\gamma = \eta = 1$, confidence floor $c_0 = 0.05$, blend strength $\beta = 0.5$, and $\epsilon = 10^{-6}$ in all settings.

GD^{2}PO baseline. The GD^{2}PO rows in Tables 1 and 2 use the hard conflict-filtering variant of GD^{2}PO (Liu et al., 2026a). Starting from GDPO's separately group-normalized advantage for each reward, GD^{2}PO-Hard removes a rollout whenever its nonzero reward-wise scalar advantages contain opposing signs. It then scales each prompt group's surviving advantages by that group's retained-rollout fraction and whitens over retained response tokens only. Reward weights are equal, and every other model, data, rollout, and optimization setting is identical to the corresponding GDPO run.

B. Why the Anchor Breaks the Teacher Ceiling

This supplementary material formalizes the claim of Section 4.3 that the two training signals of Eq. equation 9 play distinct, complementary roles. The forward-KL OPD loss can at best reproduce the teacher mixture, whereas the anchor's policy gradient is the only term that remains active at that point. Moreover, on a sparse reward, the anchor becomes informative only after OPD has lifted the student's success rate. Throughout, we treat the frozen teacher mixture of Eq. equation 5 as a fixed conditional distribution $q(\cdot \mid s)$, write d_θ for the state (prefix) distribution induced by the student's own rollouts, and ignore PPO-style clipping, which is inactive at the on-policy point where the importance ratio equals one.

Pure OPD is capped at the teacher mixture. The on-policy distillation objective of Eq. equation 7 is

$$\mathcal{L}_{\mathrm{OPD}}(\theta) = \mathbb{E}_{s\sim d_\theta}\left[\mathrm{KL}\big(q(\cdot \mid s) \,\big\|\, p_\theta(\cdot \mid s)\big)\right]. \tag{10}$$

Since $\mathrm{KL}(q \,\|\, p_\theta) \geq 0$ with equality iff $p_\theta(\cdot \mid s) = q(\cdot \mid s)$ on the support of q, any student matching the mixture on its own visited states attains the global minimum $\mathcal{L}_{\mathrm{OPD}} = 0$. Moreover, this minimum is a stationary point: the pointwise gradient is

$$\nabla_\theta \,\mathrm{KL}\left(q \,\|\, p_\theta\right) = -\sum_v q(v \mid s)\, \nabla_\theta \log p_\theta(v \mid s), \tag{11}$$

which at $p_\theta = q$ equals $-\sum_v \nabla_\theta\, p_\theta(v \mid s) = -\nabla_\theta 1 = 0$; the contribution of $\nabla_\theta d_\theta$ vanishes as well because the integrand is pointwise zero at the minimum. Gradient descent on $\mathcal{L}_{\mathrm{OPD}}$ alone therefore terminates at the mixture policy: the student inherits the teachers' behavior but receives *no* signal that would push its expected task reward above that of the mixture. This is the imitation ceiling observed in Table 3, where the OPD-only student reaches 5.544 Overall, barely above the teachers it imitates (5.500/5.522).

The anchor's gradient survives at the ceiling. The anchor is a policy-gradient term driven by the GDPO advantage $\hat{A}^{\mathrm{GDPO}}$ of Eq. equation 3:

$$-\nabla_\theta \mathcal{L}_{\mathrm{anchor}} = \mathbb{E}_{o\sim\pi_\theta}\left[\hat{A}^{\mathrm{GDPO}}(o)\, \nabla_\theta \log \pi_\theta(o)\right], \tag{12}$$

an ascent direction on the group-normalized task reward. Its stationary points are local optima of the reward, not matches to any teacher; in particular, the mixture q is not a stationary point of $\mathcal{L}_{\mathrm{anchor}}$ unless q already locally maximizes the task reward. Hence at the OPD fixed point $p_\theta = q$,

$$\nabla_\theta \mathcal{L}_{\mathrm{total}} = \nabla_\theta \mathcal{L}_{\mathrm{anchor}} + \lambda \cdot 0 = \nabla_\theta \mathcal{L}_{\mathrm{anchor}} \neq 0 \tag{13}$$

in general, so optimization of Eq. equation 9 continues *through* the imitation ceiling in the direction that increases task reward—exactly the +0.095 Overall lift of SMOPD over its anchor-free counterpart in Table 3.

Under a sparse reward, OPD is what activates the anchor. Why not rely on the anchor alone, then? For a binary reward with per-rollout success probability p_θ^{succ} under the current policy, a group of G i.i.d. rollouts receives identical rewards on that dimension with probability $(p_\theta^{\mathrm{succ}})^G + (1 - p_\theta^{\mathrm{succ}})^G$, and by Eq. equation 1 such a degenerate group contributes exactly zero advantage on that dimension. The anchor's expected signal on the sparse dimension is therefore proportional to the probability of a *mixed* group,

$$1 - (p_\theta^{\mathrm{succ}})^G - (1 - p_\theta^{\mathrm{succ}})^G \;\approx\; G\, p_\theta^{\mathrm{succ}} \quad (p_\theta^{\mathrm{succ}} \to 0), \tag{14}$$

which vanishes linearly with the success rate: near $p_\theta^{\mathrm{succ}} \approx 0$ the anchor is inert on exactly the dimension that needs it most. The OPD gradient of Eq. equation 11, by contrast, is *dense*: it is nonzero at every position where the student deviates from the mixture, independent of within-group reward variance, and the format teacher concentrates its mass precisely on the wrapper tokens the student is missing. Distillation therefore raises p_θ^{succ} rapidly; once the success rate is bounded away from 0 and 1, mixed groups occur with constant probability, the sparse dimension's advantage becomes non-degenerate, and the anchor's ascent direction of Eq. equation 12 takes over.

In summary, the two losses are complementary by construction: OPD supplies the dense, reward-independent gradient that carries the student to the teachers' level and activates the sparse dimension's group signal, while the anchor is the only term whose gradient survives at the imitation ceiling—and is thus what pushes the student *beyond* its teachers.

C. Sampled-Token Reverse-KL Estimator

Section 4.3 compares forward KL against reverse KL while keeping uniform teacher mixing, the GDPO anchor, the training recipe, and the evaluation checkpoint fixed. Because the reverse direction places the student in the first argument, we implement it in its natural on-policy, sampled-token form (Gu et al., 2024; Lu & Thinking Machines Lab, 2025). For each rollout token $y_t \sim \pi_{\mathrm{old}}$, every teacher returns the scalar log-probability of that token and the mixture is formed exactly:

$$\log q_{\mathrm{mix}}(y_t) = \mathrm{logsumexp}_m\left(\log \alpha_m + \log p_{\pi_m}(y_t)\right), \tag{15}$$

Backbone	Method	RLLA-4K test (in-domain)			Gen.
		Acc Reward	Format Pass	RLLA Mean	BFCL AST
Qwen2.5-3B	GDPO Baseline (ref)	1.753	97.5%	2.728	75.3%
	Accuracy Teacher [0.7,0.3]	1.763	94.4%	2.707	76.4%
	Format Teacher [0.3,0.7]	1.734	97.5%	2.709	73.4%
	SMOPD	**1.782**	**97.5%**	**2.757**	**76.9%**

Table 7 | *Moderate*-skew ablation on Qwen2.5-3B: teachers use softened GDPO priority profiles rather than the aggressive skew of the main results in Table 1. API-Bank was not run for these moderate checkpoints and is therefore omitted. The GDPO baseline is reproduced for reference; **bold** marks the moderate-profile SMOPD result.

where conditioning on the prefix s_t is implicit. Let $D_{\text{rev}}(\theta) := D_{\text{KL}}(\pi_\theta \| q_{\text{mix}})$. The desired reverse divergence and its score-function gradient (Williams, 1992) are

$$\begin{aligned} D_{\text{rev}}(\theta) &= \mathbb{E}_{y\sim\pi_\theta}[\log \pi_\theta(y) - \log q_{\text{mix}}(y)]\,, \\ \nabla_\theta D_{\text{rev}}(\theta) &= \mathbb{E}_{y\sim\pi_\theta}[\delta_\theta(y)\nabla_\theta \log \pi_\theta(y)]\,, \\ \delta_\theta(y) &= \log \pi_\theta(y) - \log q_{\text{mix}}(y). \end{aligned} \tag{16}$$

Accordingly, we detach the sampled log-ratio and use it as a token-level advantage in a clipped policy-ratio update:

$$\begin{aligned} k_1(y_t) &= \log \pi_{\text{old}}(y_t) - \log q_{\text{mix}}(y_t), \\ A_t &= -\,\text{stopgrad}[k_1(y_t)]\,, \\ \rho_t &= \frac{\pi_\theta(y_t)}{\pi_{\text{old}}(y_t)}, \\ \mathcal{L}_{\text{rev-PG}} &= -\mathbb{E}_t[\text{clip}(\rho_t, 1-\epsilon, 1+\epsilon)A_t]\,. \end{aligned} \tag{17}$$

At the on-policy point $\pi_\theta = \pi_{\text{old}}$, the expected gradient of Eq. equation 17 matches Eq. equation 16.

This detached policy-gradient construction is essential. Although a sampled k_3 scalar (Schulman, 2020) has the reverse-KL value in expectation, directly differentiating it while treating the rollout sampling distribution as fixed does not recover the reverse-KL gradient; it therefore cannot serve as a valid direction ablation. We exclude that invalid control and report only the sampled-token estimator in Table 4. The sampled-token reverse-KL update remains below forward KL in this setting, but the comparison does not establish that reverse KL is universally inferior.

D. Ablation: Sensitivity to the Priority Profile

Our main experiments specialize teachers with aggressive priority profiles ($[0.9, 0.1]/[0.1, 0.9]$). Here we ablate the *strength* of that skew, re-running the entire acc+format pipeline on Qwen2.5-3B with softened *moderate* weights ($[0.7, 0.3]$ accuracy-heavy, $[0.3, 0.7]$ format-heavy); Table 7 reports the results. The moderate weights yield two competitive, complementary teachers (94.4%/97.5% format), and merging works at least as well as under aggressive weights: moderate SMOPD attains 2.757—the best result among all 3B acc+format runs—beating both of its own teachers (2.707/2.709), the GDPO baseline (2.728), and the aggressive-profile student (2.738). The advantage of SMOPD is thus robust to how sharply the rewards are skewed: as long as the teachers remain complementary, merging surpasses the scalarized baseline, and the softer skew performs, if anything, marginally better.

E. Adaptive Teacher-Mixing Gates

SMOPD's default mixes teachers with *uniform* weights $\alpha_m = 1/M$ (Section 3.2). Here we ask whether a more adaptive, input-dependent gate could do better by routing more weight to the teacher that is most relevant at

each sequence or token, and we define two *confidence-failure* gates that steer the uniform prior $1/M$ using three signals derived from the teachers and rewards: each teacher's *confidence* (how peaked its top-κ distribution is), the student's *reward failure* on that teacher's target dimension, and inter-teacher *disagreement*.

Signals. At token t, teacher m's confidence combines the normalized negative entropy and the top-two margin of its (top-κ, renormalized) distribution $p_{\pi_m}(\cdot \mid s_t)$:

$$c_{m,t} = \tfrac{1}{2}\Big(1 - \tfrac{H\left[p_{\pi_m}(\cdot|s_t)\right]}{\log \kappa}\Big) + \tfrac{1}{2}\big(p^{(1)}_{\pi_m}(s_t) - p^{(2)}_{\pi_m}(s_t)\big), \tag{18}$$

where $H[\cdot]$ is Shannon entropy and $p^{(1)} \geq p^{(2)}$ are the two largest probabilities. Reward failure uses the response's reward r_m on teacher m's dimension, min–max normalized to $[0, 1]$:

$$f_m = 1 - \mathrm{clamp}\Big(\tfrac{r_m - r_{\min}}{r_{\max} - r_{\min}},\, 0,\, 1\Big), \tag{19}$$

where $[r_{\min}, r_{\max}]$ is the reward range of that dimension; a teacher is thus upweighted exactly where the student is failing its reward. Disagreement $d_t \in [0, 1]$ is the fraction of teacher pairs whose top-1 tokens differ at s_t.

Sequence-level gate. Averaging $c_{m,t}$ over the response mask gives $\bar{c}_m$, and the raw weight combines the three signals over the uniform prior:

$$\tilde{\alpha}_m = \tfrac{1}{M}\,(\bar{c}_m + c_0)^{\eta}\,(f_m + \epsilon)^{\gamma}, \quad \alpha_m = \frac{\tilde{\alpha}_m}{\sum_j \tilde{\alpha}_j}. \tag{20}$$

Averaging disagreement to $\bar{d}$, we blend back toward the prior and apply a floor ϕ so no teacher is silenced:

$$\begin{aligned} \alpha_m &\leftarrow \Phi_\phi\left[(1 - \beta\bar{d})\,\alpha_m + \beta\bar{d}\cdot\tfrac{1}{M}\right], \\ \Phi_\phi[x]_m &= \phi + (1 - \phi M)\,\tfrac{x_m}{\sum_j x_j}. \end{aligned} \tag{21}$$

Token-level gate. The same construction is applied *per position*: $c_{m,t}$ and d_t are used directly (no averaging), yielding position-specific weights $\alpha_{m,t}$ that let different teachers dominate at different tokens. Reward failure f_m remains sequence-level (one reward per response). Gate hyperparameters are fixed across all settings and listed in Sec. A.

Results. Tables 8 and 9 compare the two gates against SMOPD's uniform mixing on all five settings. The 1.5B RLLA block uses the aggressive main profile, while the 3B RLLA block uses the moderate-profile ablation of Table 7. Two findings stand out. First, **the three variants stay within a narrow band**: on RLLA Mean the spread never exceeds 0.06, and on safety Overall it never exceeds 0.02. Second, **uniform mixing is the strongest RLLA rule in both displayed profile settings**: it reaches 2.740 at 1.5B and 2.757 in the moderate 3B ablation, compared with 2.684/2.690 and 2.733/2.730 for the sequence/token gates. Token-level routing wins only on Qwen2.5-7B safe alignment (5.651 vs. 5.646), while uniform ties or leads in the remaining safety settings. The adaptive mechanisms therefore add no consistent benefit over the parameter-free mixture, motivating uniform mixing as SMOPD's default.

F. API-Bank LLM-Judge Protocol and Original Exact-Match Analysis

The main text reports API-Bank under a semantic LLM-judge metric rather than the benchmark's original strict matcher. Figure 4 summarizes the complete evaluation path, and Figure 5 shows the verbatim judge prompt. For every saved model response, we first apply the original exact matcher. Exact successes are retained as correct; only exact failures are sent to the judge (`qwen3.7-plus`, one query per failed item), together with the recent dialogue turns, the gold tool call, and the predicted call. The judge accepts a failure only when it uses the correct tool and its arguments are functionally equivalent to the requested call. Benign casing or formatting differences and harmless optional parameters may therefore be rescued, whereas wrong tools, missing required fields, unsupported values, or outcome-changing arguments remain incorrect. No model is re-run.

The final metric is $(N_{\text{exact}} + N_{\text{rescued}})/N_{\text{total}}$, computed separately for L1/L2/L3 and as micro-accuracy over all 597 items. Under this metric (Table 1), SMOPD obtains the highest Overall score at both scales: 87.10% at 1.5B and 77.39% at 3B, respectively 3.52 and 1.01 percentage points above the strongest scalarized baseline.

		RLLA-4K test (in-domain)			Generalization
Backbone	**Mixing**	**Acc Reward**	**Format Pass**	**RLLA Mean**	**BFCL AST**
1.5B	SMOPD (uniform)	1.765	**97.5%**	**2.740**	70.7%
	SMOPD (seq gate)	1.741	94.3%	2.684	**70.9%**
	SMOPD (token gate)	**1.787**	90.3%	2.690	**70.9%**
3B	SMOPD (uniform)	**1.782**	**97.5%**	**2.757**	76.9%
	SMOPD (seq gate)	1.765	96.9%	2.733	76.6%
	SMOPD (token gate)	1.755	**97.5%**	2.730	**77.1%**

Table 8 | Teacher-mixing comparison on the accuracy + format setting. Rows are SMOPD with uniform mixing (our main method) and the two confidence-failure gates. The 1.5B rows use aggressive teachers; the 3B rows use the moderate teachers from the ablation in Table 7, not the aggressive-profile 3B checkpoints in the main table. Per column and per backbone, **bold** marks the best and underline the second best among the three mixing rules (ties share the rank).

		HH-RLHF			PKU-SafeRLHF			Alpaca			Overall
Backbone	**Mixing**	U.	H.	Avg	U.	H.	Avg	U.	H.	Avg	**Avg**
Qwen2.5-3B	SMOPD (uniform)	**4.511**	5.743	5.127	**5.426**	6.869	6.147	**5.392**	6.071	**5.732**	**5.669**
	SMOPD (seq gate)	4.493	**5.776**	**5.134**	5.422	**6.881**	**6.152**	5.360	**6.079**	5.720	**5.669**
	SMOPD (token gate)	4.471	5.749	5.110	5.408	6.876	6.142	5.355	**6.079**	5.717	5.656
Qwen2.5-7B	SMOPD (uniform)	4.475	5.656	5.065	5.448	6.837	6.143	**5.421**	6.036	**5.729**	5.646
	SMOPD (seq gate)	**4.497**	5.664	5.080	5.447	6.826	6.136	5.383	5.991	5.687	5.635
	SMOPD (token gate)	4.495	**5.669**	**5.082**	**5.452**	**6.847**	**6.149**	5.402	**6.044**	5.723	**5.651**
Llama-3.2-3B	SMOPD (uniform)	**4.437**	**5.633**	**5.035**	**5.349**	**6.756**	**6.052**	5.406	5.962	5.684	**5.590**
	SMOPD (seq gate)	4.413	5.588	5.001	5.337	6.731	6.034	**5.427**	**5.974**	**5.701**	5.578
	SMOPD (token gate)	4.422	5.611	5.016	5.329	6.738	6.034	5.416	5.919	5.667	5.572

Table 9 | Teacher-mixing comparison on safe alignment. Rows are SMOPD with uniform mixing (our main method) and the two confidence-failure gates. Per column and per backbone, **bold** marks the best and underline the second best among the three mixing rules (ties share the rank).

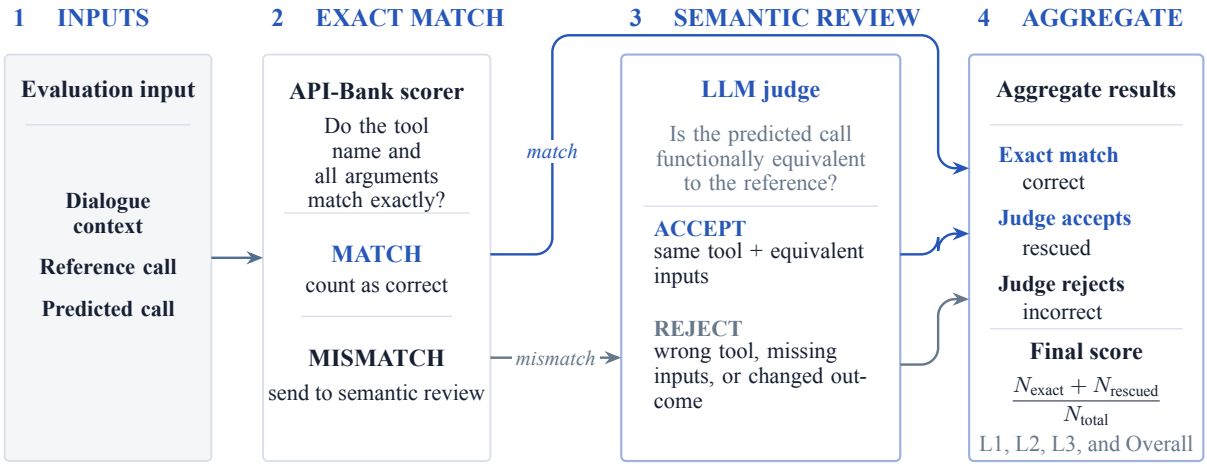


Figure 4 | API-Bank LLM-judge evaluation pipeline. The original exact matcher is applied first, and its successes remain correct. Only failed exact matches undergo semantic review using the saved dialogue, reference call, and prediction. A semantically equivalent call is rescued; a functionally different call remains incorrect. The final per-level and Overall scores combine original exact successes with judge-rescued failures.

You are a strict evaluator of tool-calling correctness for the API-Bank benchmark.
You are given a multi-turn dialogue, the GROUND-TRUTH tool call that correctly solves the user's final need, and a MODEL's predicted tool call. Decide whether the model's prediction is FUNCTIONALLY CORRECT: it must call the right tool AND supply parameters that faithfully fulfill the user's request as expressed in the dialogue, i.e. be semantically equivalent to the ground truth.
Judge CORRECT (Yes) when the prediction differs from the ground truth ONLY in ways that do not change the outcome, for example:

- Case, whitespace, or punctuation differences in parameter values.
- Synonyms / equivalent phrasings that refer to the same entity or value.
- Reasonable interpretations of time boundaries that still capture the user's stated intent (e.g. "up to March 12th" as 2023-03-12 23:59:59 vs 2023-03-12 00:00:00).
- Equivalent formatting of the same value (e.g. 5 vs "5", equivalent date formats).
- Extra optional parameters that are consistent with the dialogue and do not alter the core action.

Judge INCORRECT (No) when the prediction:

- Calls a different or wrong tool, or one that does not fulfill the request.
- Omits a required parameter, or fills a required parameter with a wrong or contradictory value.
- Uses a value that changes the meaning or outcome (different user, amount, target, time span, etc.).
- Hallucinates values not supported by the dialogue.

Only the FINAL required tool call matters. Base your decision on the user's actual need in the dialogue, not on superficial string matching. Be strict: do not pass genuinely wrong calls, but do not fail calls that are merely phrased or formatted differently.
Dialogue (most recent turns)
`{dialogue}`
Ground-truth tool call
`{gold}`
Model predicted tool call(s)
`{pred}`
Respond in EXACTLY this format and nothing else:
Reasoning: <one concise sentence>
Judgment: <Yes or No>

Figure 5 | Verbatim prompt of the API-Bank LLM judge (`qwen3.7-plus`). The placeholders `{dialogue}`, `{gold}`, and `{pred}` are filled with the recent dialogue turns, the reference tool call, and the model's predicted call(s) of each exact-match failure; the judge returns a one-sentence rationale and a binary verdict.

Item	Reference param	3B output param	Difference
L1-8	`symptom:"fatigue"`	`symptom:"Fatigue"`	casing
L1-9	`symptom:"fatigue"`	`symptom:"Chronic fatigue syndrome"`	more specific
L1-30	`content:"Meeting"`	`content:"Meeting with team"`	fuller context
L1-32	`appointment_id:"78901234"`	`..., date:"2023-08-05"`	extra field

Table 10 | Representative API-Bank items where the 3B model calls the correct tool but is scored wrong by exact name-and-parameter matching. Each 3B output is semantically correct or more complete than the reference; the failure is purely a literal-string mismatch.

Original exact-match metric. The original API-Bank scorer requires exact equality of both the function name and the *entire* parameter dictionary; any difference in casing, specificity, formatting, or additional fields counts as a failure. Table 11 preserves these strict scores for comparison with the original ToolRL/GD^{2}PO evaluation protocol. Unlike the LLM-judge results in the main text, they primarily measure literal agreement with the reference call.

Quantitative diagnosis of strict matching. On the 597 items shared by the 1.5B and 3B SMOPD runs, the exact matcher counts 392 correct for 1.5B and 347 for 3B. Of the 66 items that 1.5B passes but 3B fails, **62** are cases where 3B invokes the *correct* tool but its argument string does not match the reference literally, and only 4 are genuine refusals or clarification requests. These disagreements concentrate on the simplest Level-1 items (38 of 66), precisely where the target is a short literal string and extra model capability cannot help. The lower exact-match score of Qwen2.5-3B is therefore a literal-matching artifact rather than a regression in tool-calling ability: the more capable backbone tends to produce richer argument strings—more specific values,

Method	L1	L2	L3	Overall
Qwen2.5-1.5B				
GRPO Baseline	62.66	53.73	**48.85**	58.63
GDPO Baseline	67.42	56.72	47.33	61.81
GD^{2}PO Baseline	64.66	53.73	45.04	59.13
Accuracy Teacher [0.9,0.1]	64.91	58.21	49.62	60.80
Format Teacher [0.1,0.9]	72.18	62.69	38.17	63.65
SMOPD	**74.44**	**64.18**	39.69	**65.66**

Method	L1	L2	L3	Overall
Qwen2.5-3B				
GRPO Baseline	66.67	47.76	27.48	55.95
GDPO Baseline	65.41	**56.72**	38.17	58.46
GD^{2}PO Baseline	66.92	50.75	**41.98**	**59.63**
Accuracy Teacher [0.9,0.1]	67.67	52.24	30.53	57.79
Format Teacher [0.1,0.9]	68.67	52.24	42.75	61.14
SMOPD	**67.92**	49.25	32.82	58.12

Table 11 | Original strict API-Bank exact-match accuracy (%) on the same saved generations as Table 1. **Overall** is micro-accuracy over all 597 items (399 L1, 67 L2, and 131 L3). These are the original name-and-full-parameter-dictionary matching scores, not the semantic LLM-judge metric used in the main text. Per column and backbone, **bold** marks the best and underline the second best among the three scalarized baselines and SMOPD; single-reward teachers are shown for context and excluded from ranking.

fuller context, or extra optional fields (Table 10)—which deviate from the short reference strings more often, so the strict matcher penalizes exactly the additional capability that the semantic judge rescues.

Representative cases. Table 10 lists 3B outputs that are semantically correct—often *more* complete or faithful to the user's request—yet scored wrong by exact matching.

We use the semantic judge in the main table because it measures functional equivalence rather than literal reproduction, while retaining Table 11 for direct comparability with prior exact-match reporting. Together with BFCL, the judge results show that strict matching can understate functional tool-use ability without changing the central within-backbone comparison: at both scales, SMOPD attains the highest judged API-Bank accuracy of all compared methods.

G. Full BFCL-v4 Category Breakdown

Table 12 reports every BFCL-v4 category for the aggressive-profile models of Table 1. It shows why the full-suite *Overall* score (last column, 15–23%) is misleadingly low: it averages in categories our single-turn tool models were never trained for—*Multi-Turn* is 0% for every model, and the Web-Search/Memory categories return no score—while the models are in fact strong on the abstract-syntax-tree (AST) function-calling categories they target (76–84% non-live, 60–72% live). Our main-text **BFCL AST** metric is the mean of the Non-Live and Live AST columns. At 3B every SMOPD variant matches or exceeds the GDPO baseline's AST accuracy, whereas at 1.5B the scalarized baselines remain strongest (GD^{2}PO 73.3% vs. SMOPD 70.7%).

Size	Method	**AST**	**Non-Live AST**					**Live AST**					**Multi**	**Relevance**		**Over-**
		(ours)	Acc	Simp	Mult	Par	P.M.	Acc	Simp	Mult	Par	P.M.	**Turn**	Rel	Irrel	**all**
1.5B	GRPO Base	68.6	76.90	67.58	83.50	82.00	74.50	60.33	70.16	58.21	56.25	50.00	0.00	100.00	35.93	17.32
	GDPO Base	72.6	80.60	70.92	88.00	85.00	78.50	64.54	74.81	62.49	56.25	50.00	0.00	100.00	35.27	18.04
	GD^{2}PO Base	73.3	80.35	71.92	88.50	82.00	79.00	66.32	74.81	64.58	56.25	58.33	0.00	100.00	35.15	18.18
	Accuracy Teacher	70.5	78.83	70.83	87.50	81.00	76.00	62.10	72.09	59.73	56.25	62.50	0.00	100.00	37.44	17.84
	Format Teacher	70.5	76.92	69.17	85.50	79.50	73.50	64.03	73.26	62.01	56.25	58.33	0.00	100.00	15.72	15.67
	SMOPD	70.7	78.31	71.25	88.50	79.00	74.50	63.06	72.09	61.06	56.25	58.33	0.00	93.75	23.65	16.50
	seq gate	70.9	78.23	70.92	88.50	78.00	75.50	63.58	71.32	61.92	56.25	58.33	0.00	100.00	21.02	16.28
	token gate	70.9	78.08	71.33	88.50	78.00	74.50	63.66	74.03	61.54	50.00	54.17	0.00	100.00	20.02	16.18
3B	GRPO Base	75.8	83.00	73.50	91.50	85.00	82.00	68.54	63.18	70.28	68.75	50.00	0.00	81.25	79.50	23.10
	GDPO Base	75.3	80.56	71.25	91.00	82.50	77.50	70.02	65.89	71.04	75.00	66.67	0.00	87.50	76.80	22.74
	GD^{2}PO Base	76.5	82.69	73.75	92.50	84.00	80.50	70.32	67.83	71.04	75.00	62.50	0.00	81.25	75.57	22.86
	Accuracy Teacher	77.2	82.83	74.33	92.00	81.50	83.50	71.58	69.77	72.17	75.00	62.50	0.00	87.50	62.78	21.72
	Format Teacher	76.1	83.15	74.08	92.00	84.50	82.00	69.06	64.34	70.47	75.00	54.17	0.00	81.25	76.86	22.91
	SMOPD	77.2	83.15	73.58	92.50	84.50	82.00	71.21	68.60	72.08	75.00	58.33	0.00	87.50	73.42	22.78
	seq gate	77.3	83.71	73.83	93.00	84.50	83.50	70.84	66.67	72.17	68.75	58.33	0.00	93.75	73.23	22.78
	token gate	76.2	82.52	73.58	92.50	82.50	81.50	69.95	66.67	71.13	62.50	58.33	0.00	87.50	74.93	22.74

Table 12 | Full per-category BFCL-v4 accuracy (%) for the aggressive-profile models of Table 1. **AST (ours)** is the mean of the *Non-Live* and *Live AST* block accuracies (our main-text BFCL metric); P.M. abbreviates Parallel-Multiple. The full-suite **Overall** average is dominated by categories these single-turn tool callers never see (Multi-Turn; Web-Search and Memory return no score and are omitted), motivating the AST metric used in the main text.