LEXICOGRAPHIC MULTI-OBJECTIVE ON-POLICY DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning from verifiable rewards (RLVR) usually optimizes answer correctness, yet useful language-model behavior also requires high-quality reasoning and concise responses. Existing multi-reward post-training methods typically scalarize rewards or combine specialists without explicitly protecting a reward priority order. This is problematic when trade-offs are asymmetric: conciseness, for example, should not improve at the cost of correctness. We introduce *Lexicographic Multi-Objective On-Policy Distillation* (LMOPD), a multi-teacher method for integrating reward-specialized policies under explicit priorities. For each student rollout, LMOPD selects the specialist for the first objective whose gate detects a deficiency, then locally projects its centered log-policy correction to remove components that oppose higher-priority specialists. We evaluate 30B-A3B mixture-of-experts transformer models in two- and four-expert settings on three math benchmarks, measuring retained specialist gains. With two experts, LMOPD's point estimates fully retain the accuracy and reasoning-quality gains while acquiring 46.9% of the conciseness gain. With four experts, it retains $\approx 90\%$ of both the accuracy gain and reasoning-correctness gain, compared to only $\approx 57\%$ by the next best evaluated baseline. Matched four-expert ablations show that lexicographic routing outperforms random routing and that projection further strengthens both top-priority capabilities. Across both scales, LMOPD preserves the highest-priority capabilities more effectively than the existing baselines we evaluate, demonstrating the value of explicit priorities for specialist integration.

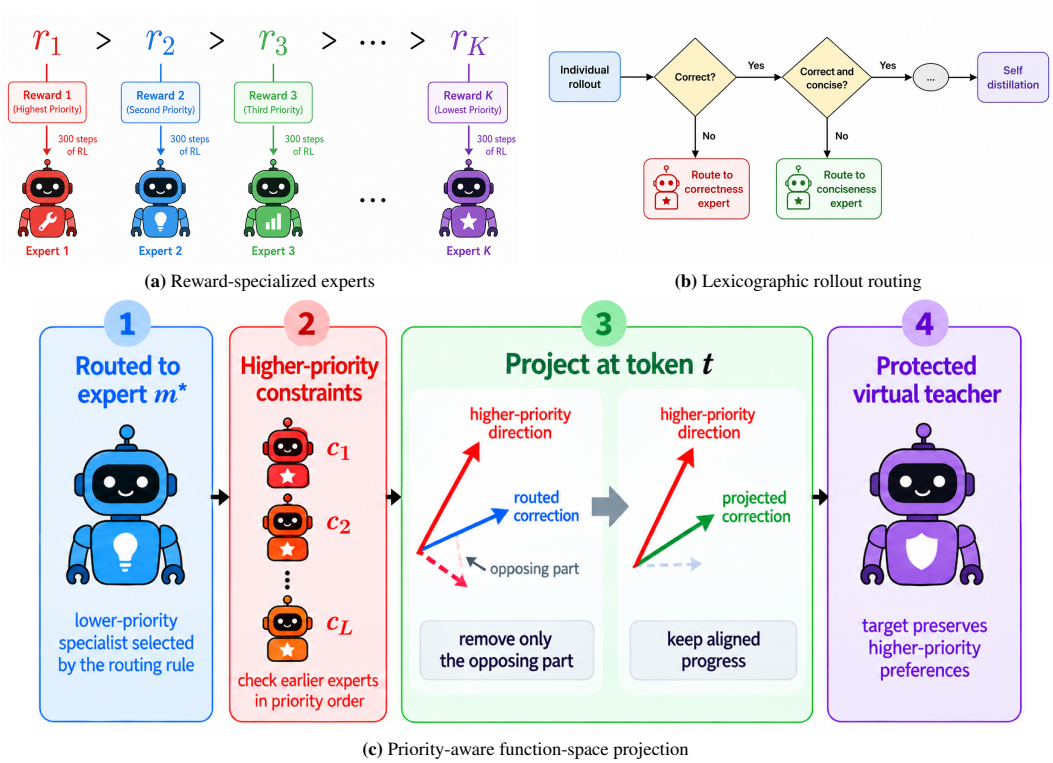
1 INTRODUCTION

Post-training has become a critical stage in building large language models (LLMs) that can solve difficult reasoning and agentic tasks. Policy-gradient methods such as Proximal Policy Optimization (PPO) (Schulman et al., 2017) and Group Relative Policy Optimization (GRPO) (Shao et al., 2024) enable models to improve from outcome-level feedback, and reinforcement learning from verifiable rewards (RLVR) has been especially effective in domains such as mathematics and code where correctness can be automatically verified. Modern reasoning systems consequently devote substantial post-training compute to reinforcement learning on verifiable tasks (Yang et al., 2025).

Outcome correctness, however, is only one dimension of response quality. A useful mathematical assistant should also reason correctly, verify its work, avoid unnecessary dead ends, and communicate concisely. These objectives can conflict and have reward signals with different scales and densities. A standard solution is scalarization; GDPO, for example, normalizes reward dimensions separately before combining them (Liu et al., 2026a). A fixed scalarization still treats objectives as compensatory: enough improvement on a lower-priority axis can offset degradation of a prerequisite. Many applications instead require a non-compensatory hierarchy in which presentation quality cannot redeem an incorrect solution.

An alternative is to specialize separate models and combine them after training. Rewarded Soups (Ramé et al., 2023), for example, interpolates parameters of models optimized for different rewards. On-policy distillation (OPD) instead transfers dense teacher feedback on student-generated trajectories (Agarwal et al., 2024). Multi-teacher OPD (MOPD) routes rollouts to domain-specialized teachers (Ma et al., 2026); there, the prompt domain determines which teacher rollouts are routed

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.



(a) Reward-specialized experts **(b)** Lexicographic rollout routing

(c) Priority-aware function-space projection

Figure 1: **LMOPD overview. (a)** We train reward-specialized experts and assign each reward to an expert; one expert may serve multiple rewards. **(b)** Each student rollout is checked in that order and routed to the first objective whose gate detects a deficiency; a rollout that passes every gate is self-distilled. **(c)** At each token, LMOPD removes only components of the routed specialist's correction that oppose earlier specialists, producing a projected virtual-teacher target.

to. We study when several objectives apply to the *same* prompt, where the missing signal is instead which requirement the individual sampled solution fails.

We propose **Lexicographic Multi-Objective On-Policy Distillation (LMOPD)** for priority-ordered specialist integration. LMOPD uses each rollout's reward profile to make the routing decision. We assume an ordered set of objectives $r_1 \succ r_2 \succ \cdots \succ r_K$ and a collection of reward-specialized experts. For each on-policy student rollout, LMOPD walks through the rewards in priority order. A discrete objective routes to its expert when its maximum acceptable reward has not been achieved; a continuous objective routes when the rollout has negative group-relative advantage on that objective. The first objective that triggers determines the teacher. If every objective passes, the rollout is routed to the student itself (effectively contributing no gradient) to avoid unnecessarily perturbing already correct behavior. Higher-priority objectives therefore have the first opportunity to correct a trajectory, while lower-priority experts only act after higher-priority requirements are satisfied.

Routing alone does not prevent a selected lower-priority expert from proposing a token-level correction that opposes a higher-priority expert. We therefore represent each expert's correction as a centered log-policy delta and sequentially remove components with negative inner product against the higher-priority expert deltas. This projection is performed in function space rather than parameter-gradient space and is implemented through streamed per-token Gram matrices, avoiding one additional student backward pass per expert.

This paper makes three contributions:

- We formulate *lexicographic corrective routing* for multi-teacher OPD. The per-rollout rule handles absolute discrete criteria and relative continuous rewards, allows several objectives to share one expert, and self-distills when no correction is required.

- We introduce a one-sided function-space projection that removes components of the routed correction that oppose higher-priority experts, together with a streamed Gram-space implementation that requires no extra student backward passes.
- We evaluate LMOPD in two- and four-expert settings at 30B scale and isolate routing and projection through component ablations.

We evaluate whether LMOPD retains prerequisite capabilities, improves teacher assignment over random routing, and redirects integration when projection activates. LMOPD is a local supervision heuristic, not a claim of globally optimal lexicographic control.

2 Related Work

RLVR and multi-reward reinforcement learning. PPO (Schulman et al., 2017) and GRPO (Shao et al., 2024) are widely used policy-gradient methods for LLM post-training. RLVR couples these updates with automatically verifiable outcome rewards, particularly for mathematical and coding tasks. Multi-reward extensions introduce additional reward dimensions for properties such as style, format, length, or process quality. GDPO (Liu et al., 2026a) identifies a failure mode in naively applying group normalization after summing heterogeneous rewards and instead normalizes each reward dimension independently before aggregation. This preserves reward-specific variation, but the final policy update still depends on a scalarized combination of objectives. Recent work also uses conditional reward structures. SpatialThinker explicitly uses lexicographically gated rewards in multimodal RL (Batra et al., 2025), while SCOPE-RL gates process-quality rewards on final-answer correctness in mathematical reasoning (Liu et al., 2026b). LMOPD shares the intuition that lower-priority feedback should not override task success, but uses this hierarchy to select a *teacher distribution* rather than to construct a scalar reward.

Combining reward-specialized policies. Rewarded Soups (Ramé et al., 2023) trains policies on different rewards from a shared initialization, linearly interpolates their weights, and uses the resulting merged model for all inputs and trajectories.

On-policy and multi-teacher distillation. Knowledge distillation trains a student to match a teacher distribution; reverse KL provides a mode-seeking signal for autoregressive LLMs (Gu et al., 2024). GKD uses student-generated sequences (Agarwal et al., 2024) instead, providing feedback that is both dense and on-policy. MOPD extends GKD by dispatching rollouts to domain-specialized teachers to mix capabilities across domains and multiple teachers(Ma et al., 2026). MOPD has emerged as a key component of frontier LLM post-training recipes, including MiMo-V2-Flash Xiao et al. (2026), DeepSeek-V4 Xu et al. (2026), GLM-5 Zeng et al. (2026), and Nemotron-Cascade 2 Yang et al. (2026). SMOPD, a concurrent work, first trains reward specialists, then merges a token-level mixture of all teachers with a balanced multi-reward policy-gradient anchor (Wang et al., 2026).

At the mechanism level, LMOPD is related to PCGrad, which removes pairwise conflicting parameter-gradient components in multi-task learning (Yu et al., 2020). LMOPD instead filters one routed teacher correction only against earlier priorities, using an unweighted token-level function-space surrogate rather than parameter gradients.

Lexicographic optimization. Lexicographic multi-objective reinforcement learning optimizes objectives according to an ordered preference relation rather than treating all trade-offs as interchangeable. Recent work studies this setting theoretically and algorithmically, including regret guarantees for linear MDPs (Xue et al., 2025) and thresholded lexicographic formulations (Tercan & Prabhu, 2024). LMOPD does not attempt to provide the same policy-level optimality guarantees. Instead, it borrows the priority structure to select a corrective teacher signal and locally project that signal during distillation.

Across these lines of work, LMOPD differs in three coupled choices: unlike domain-routed MOPD, it routes individual rollouts by reward deficiency; unlike Rewarded Soups and SMOPD, it does not globally or simultaneously merge all specialists; and unlike GDPO, it does not collapse priorities into a scalarized advantage.

3 METHOD

LMOPD has two conceptually separable components. First, *lexicographic rollout routing* uses the reward profile of each student-generated completion to select the specialist associated with its highest-priority deficiency. Second, *priority-aware projection* modifies the selected specialist's token-level correction whenever it opposes a higher-priority specialist. This decomposition yields a routing-only method and a routing-plus-projection method with the same rollout assignments and distillation loss. Figure 1 summarizes the complete pipeline.

3.1 LEXICOGRAPHIC MULTI-OBJECTIVE SETTING

Let $x \sim \mathcal{D}$ denote a prompt and let a student policy π_θ generate a completion $y = (y_1, \ldots, y_T) \sim \pi_\theta(\cdot \mid x)$. We evaluate a rollout with K reward functions

$$\mathbf{r}(x, y) = (r_1(x, y), \ldots, r_K(x, y)), \tag{1}$$

with priority ordering $r_1 \succ r_2 \succ \cdots \succ r_K$. For a policy π, define performance on objective i as

$$J_i(\pi) = \mathbb{E}_{x\sim\mathcal{D},\, y\sim\pi(\cdot|x)}[r_i(x, y)]. \tag{2}$$

The ordering encodes an asymmetric preference: improvement on a lower-priority objective is not exchangeable for degradation of a higher-priority objective.

We are additionally given M frozen specialists $\{\pi_{\phi_1}, \ldots, \pi_{\phi_M}\}$. Let $e(i) \in \{1, \ldots, M\}$ map reward i to the expert used to correct that objective. The mapping need not be one-to-one: if one specialist is strong on several correlated rewards, multiple i may share the same $e(i)$.

3.2 ON-POLICY DISTILLATION

At the start of an update, we freeze an anchor $\bar{\pi}_\theta \leftarrow \pi_\theta$, sample student rollouts, and minimize a token-averaged reverse KL toward a fixed target distribution together with a hard-label cross entropy term (Agarwal et al., 2024; Hinton et al., 2015). The target is the routed specialist, the anchor itself, or the projected virtual teacher introduced below. Unlike domain-based MOPD (Ma et al., 2026), LMOPD chooses this target from the rollout's reward profile. Appendix A.1 gives the complete objective and notation.

3.3 STAGE I: LEXICOGRAPHIC ROLLOUT ROUTING

We sample G rollouts for each prompt from the frozen anchor, as in group-based RL methods:

$$y_{b,g} \sim \bar{\pi}_\theta(\cdot \mid x_b), \qquad b \in \{1, \ldots, B\}, \quad g \in \{1, \ldots, G\}. \tag{3}$$

Writing $r_{b,g,i} = r_i(x_b, y_{b,g})$, we compute the within-prompt moments and group-relative advantage:

$$\mu_{b,i} = \frac{1}{G}\sum_g r_{b,g,i}, \qquad \sigma_{b,i} = \sqrt{\frac{1}{G}\sum_g (r_{b,g,i} - \mu_{b,i})^2}, \qquad A_{b,g,i} = \frac{r_{b,g,i} - \mu_{b,i}}{\sigma_{b,i} + \epsilon}. \tag{4}$$

A single satisfaction rule is inappropriate for every reward type. For discrete rewards with a known maximum acceptable value $r_i^{\max}$, we use an absolute gate; for continuous rewards, we use relative underperformance:

$$g_i(x_b, y_{b,g}) = \mathbb{I}[r_{b,g,i} < r_i^{\max}] \text{ (discrete)}, \qquad g_i(x_b, y_{b,g}) = \mathbb{I}[A_{b,g,i} < 0] \text{ (continuous)}. \tag{5}$$

The absolute gate is important for sparse binary rewards: if every rollout in a group is incorrect, all group-relative advantages are zero, yet every rollout still requires correctness supervision.

LMOPD selects the first objective whose gate fires:

$$k^*(x_b, y_{b,g}) = \min\{i \in \{1, \ldots, K\} : g_i(x_b, y_{b,g}) = 1\}. \tag{6}$$

If the set is nonempty, the routed expert is $m^* = e(k^*)$. If every gate passes, no expert is selected and the rollout is self-distilled. Let $\beta_{m^*} \in (0, 1]$ be the routed-teacher coefficient used by the existing

OPD logit pool; $\beta_{m^*} = 1$ recovers a direct specialist target. We use $\beta_{m^*} = 0.9$ for all experiments. Writing $p_{0,b,g,t} = \bar{\pi}_\theta(\cdot \mid s_{b,g,t})$ and $p_{m,b,g,t} = \pi_{\phi_m}(\cdot \mid s_{b,g,t})$, the routing-only target is

$$q^{\text{route}}_{b,g,t} = \begin{cases} \text{softmax}((1-\beta_{m^*}) \log p_{0,b,g,t} + \beta_{m^*} \log p_{m^*,b,g,t}), & \text{if at least one gate fires,} \\ p_{0,b,g,t}, & \text{otherwise.} \end{cases} \tag{7}$$

Substituting Eq. 7 into Eq. 13 gives the routing-only LMOPD ablation. The ordering therefore gives each objective a "first right of refusal" over the rollout: incorrect responses receive capability supervision before lower-priority properties such as conciseness are considered. For projection, routing also records the ordered set $\mathcal{C}(k^*)$ of distinct higher-priority experts, excluding the routed expert itself. Appendix A.2 gives the exact construction, edge cases, and pseudocode.

3.4 Stage II: Priority-Aware Function-Space Projection

Why routing alone is insufficient. Routing controls which expert supervises a rollout, but it does not control the direction of that expert's update. A lower-priority expert can therefore assign a correction that locally opposes a higher-priority capability. LMOPD tests compatibility with an inner product between token-level corrections. The test concerns only the teacher-dependent reverse-KL component and excludes the hard-label cross-entropy term. This surrogate avoids an additional student backward pass per expert compared to computing the inner product on the actual gradients; more details can be found in Appendix A.3.

Centered log-policy corrections. For a token prefix s_t, write the frozen student anchor as $p_{0,t} = \bar{\pi}_\theta(\cdot \mid s_t)$ and expert m as $p_{m,t} = \pi_{\phi_m}(\cdot \mid s_t)$. For every vocabulary item a, define

$$\Delta_{m,t}(a) = \log p_{m,t}(a) - \log p_{0,t}(a), \qquad \widetilde{\Delta}_{m,t}(a) = \Delta_{m,t}(a) - \mathbb{E}_{a' \sim p_{0,t}}[\Delta_{m,t}(a')]. \tag{8}$$

$\widetilde{\Delta}_{m,t}(a)$ is just a centered version of $\Delta_{m,t}(a)$.

One-sided priority-aware projection. Suppose the rollout is routed to m^* and $\mathcal{C}(k^*) = (c_1, \ldots, c_L)$. We initialize $d_t^{(0)} = \beta_{m^*} \widetilde{\Delta}_{m^*,t}$, using the same routed-teacher coefficient as Eq. 7. We then visit the constraint experts once in priority order. Step ℓ is

$$u_{\ell,t} = \left\langle d_t^{(\ell-1)}, \widetilde{\Delta}_{c_\ell,t} \right\rangle, \qquad d_t^{(\ell)} = d_t^{(\ell-1)} - \frac{\min(u_{\ell,t}, 0)}{\|\widetilde{\Delta}_{c_\ell,t}\|_2^2 + \epsilon} \widetilde{\Delta}_{c_\ell,t}. \tag{9}$$

Each step removes only an opposing component, but a single sweep may leave conflicts among non-orthogonal constraints. This is a token-level surrogate, not a parameter-gradient or reward-level guarantee; Appendix A.3 gives the exact limiting cases. The projected virtual teacher is

$$q^{\text{proj}}_{b,g,t} = \text{softmax}\left(\log p_{0,t} + \tau d_t^{(L)}\right), \tag{10}$$

where the projection strength is $\tau = 1$ in our experiments. The complete method substitutes Eq. 10 into Eq. 13, with the target and projection coefficients detached from the student gradient. Appendix A.3 gives the complete algorithm, limiting cases, and one-sweep feasibility analysis. Appendix B describes the streamed Gram-space implementation and numerical safeguards.

4 Experiments

4.1 Training

Datasets. We use a proprietary mathematical-reasoning corpus for post-training.

Reward functions. We use binary answer correctness, reasoning quality judged by a 31B Gemma 4 judge (Team et al., 2026) as an average over four axes (reasoning correctness, verification, clarity, and efficiency), and a bounded length-based score S_{conc}. All rewards are in the interval $[0, 1]$. Every reported Conc metric is computed from raw S_{conc}. The conciseness specialist is trained with a correctness-gated version where it only receives S_{conc} on correct rollouts and 0 otherwise, whereas GDPO directly optimizes raw S_{conc}. Appendix C.5 gives the complete definitions.

The orders encode prerequisites rather than interchangeable utilities. Correctness comes first because presentation cannot redeem a wrong answer; reasoning correctness follows as a core capability, then verification, efficiency, and conciseness refine otherwise correct solutions. We therefore read results from highest to lowest priority and treat their macro-average as a descriptive summary.

Expert models. We train reward-specialized experts from the shared base and select teachers by target-reward validation performance. This produces the two integration settings below; See Appendix C.6 for further details on specialist training and expert selection.

Two-expert setting (three rewards).

The first setting asks whether a student can acquire a lower-priority behavior from one specialist while retaining the core mathematical capability of another. We order the objectives as $R_C \succ R_{\mathrm{RQ}} \succ R_{\mathrm{conc}}$ and use two teachers:

1. a capability expert, selected from the reasoning-specialized runs and used for both the correctness and aggregate reasoning-quality gates; and
2. a conciseness expert, used for the conciseness gate.

Because both higher-priority gates share one capability teacher, this setting isolates adding conciseness while retaining core mathematical capability. As we use one teacher for both R_C and R_{RQ}, rollouts that route to either one of those gates would get the same loss.

Four-expert setting (five rewards).

The second setting tests a longer priority chain with supervision distributed across more specialists. We use $R_C \succ R_{\mathrm{RC}} \succ R_{\mathrm{RV}} \succ R_{\mathrm{RE}} \succ R_{\mathrm{conc}}$. The four teachers are:

1. a reasoning-correctness expert, shared by the correctness and reasoning-correctness gates;
2. a reasoning-verification expert;
3. a reasoning-efficiency expert; and
4. a conciseness expert.

This setting tests integration under a longer chain in which distinct specialist corrections compete and later teachers act only after earlier gates pass. It therefore tests both multi-expert capability integration and the scaling behavior of the lexicographic routing rule.

Student model. The LMOPD student and all RL-trained specialist and baseline policies start from the same roughly 30B-parameter mixture-of-experts checkpoint, with approximately 3B active parameters per token. The model architecture is closely related to a subsequently released open-weight model; additional details will be provided in the de-anonymized version.

Baselines. We compare LMOPD with baselines that isolate different components of the method.

Rewarded Soup. Rewarded Soups (Ramé et al., 2023) linearly interpolates the parameters of reward-specialized policies trained from the same initialization. We evaluate both a uniform interpolation and a correctness-emphasized interpolation selected without using held-out test sets. This tests whether capability integration can be achieved by a static weight-space merge rather than by on-policy teacher supervision.

GDPO. GDPO (Liu et al., 2026a) is our primary multi-objective RL baseline. It optimizes raw correctness, aggregate reasoning quality, and raw S_{conc}, normalizing each reward dimension before combining the advantages into one policy-gradient direction. GDPO uses objective weights, rather than hand-designed gates, to control the trade-off. We evaluate equal weights and correctness-emphasized $5{:}1$ and $25{:}1$ weightings; the main tables show equal weights and the best-tuned $25{:}1$ run, while Table 4 retains the intermediate configuration.

Correctness-Gated RLVR. As a lexicographic reward-design control, Correctness-Gated RLVR makes correctness a prerequisite by zeroing the aggregate reasoning-quality and conciseness rewards on incorrect rollouts, then scalarizes the rewards that remain. Appendix C.7 gives both baseline objectives. Because R_{RQ} is the mean of the same four reasoning-quality axes, Correctness-Gated RLVR and each GDPO weighting receive the same three aggregate rewards in the two- and

four-expert settings. We therefore reuse the same step-300 policy checkpoints in both tables; the four-expert table only decomposes aggregate reasoning quality into component axes for evaluation.

Individual specialists. We report the frozen reward-specialized experts themselves. These models are not unified multi-objective solutions; instead, they establish the per-objective headroom available from specialization and define the reference point for retained gain.

Component ablations. The routing-only ablation is exactly the same as LMOPD but skips the projection step. This isolates the contribution of priority-aware projection. We also present a random-routing control which is exactly the same as the routing-only variant but matches experts to rollouts at random. This tests whether improvements come merely from exposure to multiple teachers or from matching teacher identity to the rollout's reward deficiencies and priority order.

Training settings. Two- and four-expert LMOPD variants train for 40 and 80 steps, respectively, and use their final checkpoints. All RL-trained specialist and baseline policies use their final step-300 checkpoint. See Appendix C for further details.

4.2 Evaluation

Datasets and protocol. We generate 50 completions per problem on AIME 2024, AIME 2025, and HMMT February 2025 and report the average and standard error of the mean (SEM) None of these benchmarks is used for training or checkpoint selection. See Appendix C for further details.

Metrics and uncertainty. We aggregate each reported objective across the three benchmarks. We report pass@1, aggregate reasoning quality (plus the three relevant component axes), and raw S_{conc} independent of correctness. Raw objective values are means with standard errors of the mean (SEM), while retained-gain SEs use the delta method.

For objective i, let $J_i(\pi)$ denote the model's mean performance, let π_0 be the shared base policy, and let $\pi_i^\star$ be the best available specialist for that objective. We report the *retained gain*

$$G_i(\pi) = \frac{J_i(\pi) - J_i(\pi_0)}{J_i(\pi_i^\star) - J_i(\pi_0)} \times 100\%. \tag{11}$$

Retained gain measures how much of the improvement created by specialization survives after capabilities are integrated into one model. A value of 0% means that the unified method does not improve over the base on that objective, while 100% means that it closes the entire base-to-specialist gap; values above 100% indicate that the integrated policy surpasses the best specialist, and negative values indicate degradation relative to the base. This normalization is useful because reward axes can differ substantially in both scale and available headroom. We report the corresponding raw objective values in Table 4 and treat the macro-average retained gain as a descriptive summary rather than a substitute for inspecting high-priority objectives individually.

5 Results

5.1 Two-Expert Retained Gain

Table 1 reports retained gain aggregated over AIME 2024, AIME 2025, and HMMT February 2025, with 50 sampled completions per problem. LMOPD preserves the two high-priority capabilities: its point estimates retain $(102.9 \pm 15.5)\%$ of the accuracy gain (raw: $0.711 \rightarrow 0.745$) and $(103.3 \pm 5.7)\%$ of the aggregate reasoning-quality gain (raw: $0.719 \rightarrow 0.772$), while retaining $(46.9 \pm 2.2)\%$ of the conciseness gain. Because 100% denotes the best-specialist endpoint, both high-priority estimates are consistent within their SEs with specialist-level performance.

LMOPD has the highest performance among unified methods (84.4%), 17.3 points above the strongest non-LMOPD baseline, the correctness-emphasized Rewarded Soup (67.1%). That soup retains substantial gains in aggregate reasoning quality and conciseness but only 72.7% of the available accuracy gain. Equal-weight GDPO reduces mean completion length from 16.3k tokens for the base to 6.2k while lowering raw accuracy from 0.711 to 0.559. At $25{:}1$, GDPO avoids this collapse: mean length rises to 13.3k tokens, and raw accuracy and aggregate reasoning quality recover to

Table 1: Two-expert retained gain. LMOPD has the highest average among unified methods while retaining gains in accuracy and aggregate reasoning quality.

Model	pass@1	RQ	Conc	Average
Weight-space merging				
Rewarded Soup (uniform; 1:1)	$61.2 \pm 13.7\%$	$61.6 \pm 5.0\%$	$67.2 \pm 2.3\%$	63.3%
Rewarded Soup (correctness-emphasized; 2:1)	$\underline{72.7 \pm 14.0\%}$	$\underline{84.3 \pm 5.3\%}$	$44.4 \pm 2.1\%$	$\underline{67.1\%}$
Multi-reward reinforcement learning				
Correctness-Gated RLVR (1,1,1)	$70.0 \pm 13.8\%$	$40.2 \pm 5.0\%$	$49.1 \pm 2.2\%$	53.1%
GDPO (1,1,1)	$-457.9 \pm 85.2\%$	$-154.4 \pm 15.3\%$	$\mathbf{355.8 \pm 9.8\%}$	-85.5%
GDPO (25,1,1)	$-31.5 \pm 20.1\%$	$-2.4 \pm 4.7\%$	$\underline{90.5 \pm 3.7\%}$	18.9%
Ours				
LMOPD	$\mathbf{102.9 \pm 15.5\%}$	$\mathbf{103.3 \pm 5.7\%}$	$46.9 \pm 2.2\%$	**84.4%**
Reference policies (not unified)				
Base	0%	0%	0%	0%
Reasoning Quality + Correctness Expert	100%	100%	$39.2 \pm 2.2\%$	79.7%
Conciseness Expert	$0.8 \pm 16.1\%$	$-20.4 \pm 6.5\%$	100%	26.8%

Bold and underlined entries are the best and second-best unified methods in each column; reference policies are excluded. LMOPD denotes routing plus priority-aware projection and uses the fixed step-40 checkpoint. Best specialist per axis: pass@1 = Reasoning Quality + Correctness Expert (0.7440), RQ = Reasoning Quality + Correctness Expert (0.7698), and Conc = Conciseness Expert (0.8131). The shared base defines 0%.

0.700 and 0.717, respectively, while retaining 90.5% of the conciseness gain. The equal-weight run is therefore consistent with length-dominant reward hacking, while stronger correctness weighting substantially mitigates it.

For qualitative analysis of LMOPD's completions, we present a case study of an individual eval problem in Appendix D.

5.2 Four-Expert Retained Gain

The four-expert setting tests whether high-priority capabilities remain protected as the priority chain grows to five objectives. Table 2 shows that LMOPD retains $(90.6 \pm 15.0)\%$ of the accuracy gain (raw: $0.711 \rightarrow 0.748$) and $(88.6 \pm 7.0)\%$ of the reasoning-correctness gain (raw: $0.755 \rightarrow 0.805$). Every other unified method reaches at most 57.1% and 63.5%, respectively. LMOPD also has the highest macro-average point estimate (39.6%, versus 34.4% for Correctness-Gated RLVR).

Because the objectives are lexicographically ordered, the columns should be read in priority order rather than as equally consequential targets. The central scaling result is therefore LMOPD's large margin on accuracy and reasoning correctness, followed by a positive composite-conciseness gain ($28.0 \pm 1.1\%$). Reasoning verification and efficiency are lower-priority objectives: their retained gains of -6.8% and -2.4% do not outweigh preservation of the first two objectives under the stated preference order. This profile is consistent with the intended behavior of LMOPD—preserving high-priority capabilities even when all specialist directions cannot be improved simultaneously. **LMOPD outperforms other multi-objective optimization baselines on important objectives.**

5.3 Component Ablations

Table 3 separates teacher assignment from priority-aware projection. Lexicographic routing's average retained-gain exceeds random routing by 28.7 points with two experts and 23.0 points with four. In the four-expert setting, random routing improves later objectives but falls below the base on accuracy and reasoning correctness; teacher exposure alone is therefore insufficient.

Projection exposure differs sharply by setting. In the two-expert run, only 0.6% of rows carry a higher-priority constraint, corresponding to 0.003–0.005 constraints per token and a projected-to-original norm ratio of 0.9989. Projection is therefore effectively inert there, so the performance shifts from routing only to full LMOPD are modest relative to their uncertainty. In the four-expert run, projection is materially engaged: it processes 0.26 constraints per token, adds constraint forwards for 13.7% of rows, and yields a norm ratio of 0.91. At the matched step-80 checkpoint, projection shifts the retained-gain point estimates for accuracy and reasoning correctness from 54.0% to 90.6% and from 69.5% to 88.6%, respectively, while reducing later-axis gains.

Table 2: Four-expert retained gain, with objectives ordered by priority. LMOPD leads unified methods on composite pass@1, RQ-corr, and average.

Model	Composite pass@1	RQ-corr	RQ-verif	RQ-eff	Composite Conc	Average
Weight-space merging						
Rewarded Soup (uniform; 1:1:1:1)	$31.4 \pm 14.3\%$	$51.6 \pm 7.5\%$	$-2.3 \pm 2.5\%$	$28.7 \pm 2.1\%$	$40.2 \pm 1.4\%$	29.9%
Rewarded Soup (correctness-emphasized; 2:1:1:1)	$30.5 \pm 11.6\%$	$\underline{63.5 \pm 10.4}\%$	$\mathbf{4.2 \pm 2.9}\%$	$19.2 \pm 2.0\%$	$28.5 \pm 0.8\%$	29.2%
Multi-reward reinforcement learning						
Correctness-Gated RLVR (1,1,1)	$\underline{57.1 \pm 9.5}\%$	$55.7 \pm 7.7\%$	$\underline{3.2 \pm 2.4}\%$	$30.8 \pm 4.4\%$	$25.2 \pm 1.8\%$	$\underline{34.4}\%$
GDPO (1,1,1)	$-373.4 \pm 55.7\%$	$-265.0 \pm 35.7\%$	$-161.5 \pm 6.1\%$	$\mathbf{95.1 \pm 2.0}\%$	$\mathbf{182.1 \pm 3.0}\%$	-104.6%
GDPO (25,1,1)	$-25.7 \pm 14.0\%$	$5.6 \pm 9.1\%$	$-35.6 \pm 3.6\%$	$\underline{36.0 \pm 2.4}\%$	$\underline{46.3 \pm 1.9}\%$	5.3%
Ours						
LMOPD	$\mathbf{90.6 \pm 15.0}\%$	$\mathbf{88.6 \pm 7.0}\%$	$-6.8 \pm 2.7\%$	$-2.4 \pm 2.8\%$	$28.0 \pm 1.1\%$	**39.6%**
Reference policies (not unified)						
Base	0%	0%	0%	0%	0%	0%
RQ-Correctness Expert	100%	100%	$7.7 \pm 3.2\%$	$-8.9 \pm 3.2\%$	$13.0 \pm 0.9\%$	42.4%
RQ-Verification Expert	$79.7 \pm 12.5\%$	$124.4 \pm 9.5\%$	100%	$-58.5 \pm 4.4\%$	$-44.3 \pm 2.1\%$	40.3%
RQ-Efficiency Expert	$-202.8 \pm 33.7\%$	$-118.5 \pm 20.7\%$	$-106.5 \pm 4.9\%$	100%	100%	-45.6%
Conciseness Expert	$0.7 \pm 12.8\%$	$25.3 \pm 9.9\%$	$-47.6 \pm 3.3\%$	$21.7 \pm 2.1\%$	$51.2 \pm 1.2\%$	10.2%

Bold and underlined entries are the best and second-best unified methods in each column; reference policies are excluded. LMOPD denotes routing plus priority-aware projection. Best specialist per axis: pass@1 = RQ-Correctness Expert (0.7515), RQ-corr = RQ-Correctness Expert (0.8117), RQ-verif = RQ-Verification Expert (0.7902), RQ-eff = RQ-Efficiency Expert (0.7827), and Conc = RQ-Efficiency Expert (0.8402).

Table 3: Routing and projection ablations. Lexicographic routing has higher performance than random routing; matched four-expert projection shifts retention toward the top priorities.

(a) Two experts

Variant	pass@1	RQ	Conc	Average
LMOPD	$102.9 \pm 15.5\%$	$103.3 \pm 5.7\%$	$46.9 \pm 2.2\%$	84.4%
LMOPD (routing only)	$95.3 \pm 12.5\%$	$97.3 \pm 4.8\%$	$43.5 \pm 1.8\%$	78.7%
LMOPD (random routing)	$30.4 \pm 14.0\%$	$46.8 \pm 4.2\%$	$72.8 \pm 2.0\%$	50.0%

(b) Four experts

Variant	Composite pass@1	RQ-corr	RQ-verif	RQ-eff	Composite Conc	Average
LMOPD	$90.6 \pm 15.0\%$	$88.6 \pm 7.0\%$	$-6.8 \pm 2.7\%$	$-2.4 \pm 2.8\%$	$28.0 \pm 1.1\%$	39.6%
LMOPD (routing only)	$54.0 \pm 13.2\%$	$69.5 \pm 11.1\%$	$-18.4 \pm 2.9\%$	$4.3 \pm 3.4\%$	$39.7 \pm 1.0\%$	29.8%
LMOPD (random routing)	$-53.7 \pm 14.2\%$	$-2.0 \pm 13.6\%$	$-39.6 \pm 3.6\%$	$53.8 \pm 2.7\%$	$75.6 \pm 1.6\%$	6.8%

6 LIMITATIONS

LMOPD serves as a local heuristic rather than global lexicographic optimality: its function-space test approximates parameter-gradient compatibility, and one sweep may leave residual conflicts.

Strict routing can starve later experts (Appendix A.4), and outcomes ultimately depend on reward and teacher quality. We evaluate only proprietary-model mathematical reasoning; open-weight and cross-domain validation remain future work. Because 30B-scale post-training is costly, multi-seed replication is limited to two-expert LMOPD: three 40 step runs (see Appendix C.4). The main-table run is the median seed by average retained gain (84.4%; the other seeds obtain 81.2% and 95.1%); Table 5 and fig. 2 give the complete results. Except for this replication study, reported SEs quantify evaluation sampling, not training-run variation.

7 CONCLUSION

We introduced LMOPD, which routes each rollout to the specialist for its highest-priority unmet objective and projects that specialist's token-level correction against earlier experts. With two experts, LMOPD retains specialist-level accuracy and aggregate reasoning quality while acquiring conciseness; with four, it retains 90.6% of the accuracy gain and 88.6% of the gain in reasoning correctness, with higher point estimates than multi-objective baselines on both top priorities. Ablations show that deficiency-aware routing is essential and that, when materially engaged with four experts, projection shifts point estimates toward these priorities. Together, these results position rollout-conditioned distillation as a practical path to integrating specialists without collapsing prioritized capabilities into a single global trade-off.

AI USE STATEMENT

In this work, we used generative AI tools for coding assistance in implementing methods, designing and providing feedback on methods and experiments, refining hypotheses, and supporting qualitative data analysis. We have not used generative AI tools for interpreting results, cleaning and reformatting datasets, and generating synthetic datasets, developing theoretical models or frameworks, formulating mathematical claims, providing critical ingredients for proving mathematical claims, writing of proofs, translation are not applicable to this work. Additionally, we used generative AI tools for modifying scientific figures, editing the research paper, analyzing existing literature. We have reviewed all AI-assisted work. LLM-generated code, experiment feedback, and paper modifications were verified by the main author for correctness. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The LMOPD routing rule is given in Eqs. 4–7 and Algorithm 1; the priority-aware target is specified in Eqs. 8–10, Algorithm 2, and the Gram-space implementation in Appendix B; and the distillation objective is specified in Eqs. 13–15. The experimental setup, baselines, and retained-gain metric are summarized in Section 4, with reward definitions, training schedules, batch construction, and evaluation details in Appendix C. The base checkpoint and training corpus used in our experiments are proprietary, so those exact assets cannot be released. The method itself requires only a student policy, white-box access to reward-specialized teacher distributions, and the stated reward functions, and can therefore be reproduced with open-weight models by training specialists from a common initialization. The paper provides implementation-level pseudocode and configuration details sufficient to reproduce the algorithm independently of the proprietary checkpoint.

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*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/5be69a584901a26c521c2b51e40a4c20-Abstract-Conference.html.

Mislav Balunović, Jasper Dekoninck, Ivo Petrov, Nikola Jovanović, and Martin Vechev. MathArena: Evaluating LLMs on uncontaminated math competitions. *arXiv preprint arXiv:2505.23281*, 2025. URL https://arxiv.org/abs/2505.23281.

Hunar Batra, Haoqin Tu, Hardy Chen, Yuanze Lin, Cihang Xie, and Ronald Clark. SpatialThinker: Reinforcing 3d reasoning in multimodal LLMs via spatial rewards, 2025. URL https://arxiv.org/abs/2511.07403.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. MiniLLM: Knowledge distillation of large language models. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/8ac015d409635f196f9e3e9dcfb9a94e-Abstract-Conference.html.

Harvard-MIT Mathematics Tournament. February 2025 tournament archive. HMMT Archive, 2025. URL https://www.hmmt.org/www/archive/282.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

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, 2026a. URL https://arxiv.org/abs/2601.05242.

Xiaojian Liu, Han Xu, Jianqiang Xia, Zhixuan Li, Ke Xu, Yiwei Dai, Xinran Chen, Changwo Wu, and Yuchen Li. SCOPE-RL: Optimizing reasoning paths before and after success, 2026b. URL https://arxiv.org/abs/2607.11506.

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, 2026. URL https://arxiv.org/abs/2606.30406.

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. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/e12a3b98b67e8395f639fde4c2b03168-Abstract-Conference.html.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. URL https://arxiv.org/abs/1707.06347.

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. URL https://arxiv.org/abs/2402.03300.

Gemma Team, Sherif El Abd, Vaibhav Aggarwal, Robin Algayres, Alek Andreev, Olivier Bachem, Ian Ballantyne, Cormac Brick, Victor Cărbune, Michelle Casbon, Mayank Chaturvedi, Aditya Chawla, Victor Cotruta, Alice Coucke, Phil Culliton, Robert Dadashi, Lucas Dixon, Mohamed Elhawaty, Utku Evci, Clément Farabet, Johan Ferret, Filippo Galgani, Sertan Girgin, Jean-Bastien Grill, Maarten Grootendorst, Jiaxian Guo, Cassidy Hardin, Yanzhang He, Steven M. Hernandez, Omri Homburger, Léonard Hussenot, Juyeong Ji, Armand Joulin, Aishwarya Kamath, Parnian Kassraie, Olivier Lacombe, Preethi Lahoti, Gaël Liu, Gus Martins, Luciano Martins, Tatiana Matejovicova, Ramona Merhej, Nikola Momchev, Sneha Mondal, Ryan Mullins, Sindhu Raghuram Panyam, Shreya Pathak, Sarah Perrin, André Susano Pinto, Etienne Pot, Angéline Pouget, Alexandre Ramé, Sabela Ramos, Douglas Reid, David Rim, Morgane Rivière, Karsten Roth, Louis Rouillard, Omar Sanseviero, Pier Giuseppe Sessa, Shane Settle, Danila Sinopalnikov, Sara Smoot, Piotr Stanczyk, Andreas Steiner, Lawrence Stewart, Ilya Tolstikhin, Michael Tschannen, Anton Tsitsulin, Nino Vieillard, Renjie Wu, Pingmei Xu, Haichuan Yang, Edouard Yvinec, Biao Zhang, Li Zhang, Joe Zou, Nicolas Aagnes, Abdelrahman Abdelhamed, Jakub Adamek, Shivani Agrawal, Shubham Agrawal, Ibrahim Alabdulmohsin, Jean Baptiste Alayrac, Uri Alon, Chandramouli Amarnath, Ankesh Anand, Chrysovalantis Anastasiou, Setareh Ariafar, François-Xavier Aubet, Kyriakos Axiotis, Federico Barbero, Joelle Barral, Alexei Bendebury, Urs Bergmann, Stanley Bileschi, Kat Black, Mathieu Blondel, Sebastian Borgeaud, Arthur Bražinskas, Ryan Burnell, Robert Busa-Fekete, Mu Cai, Daniele Calandriello, Glenn Cameron, Charlotte Caucheteux, Rahma Chaabouni, Garima Chadha, Jetha Chan, Blake Jianhang Chen, Jesse Chen, Lin Chen, Xu Chen, Derek Cheng, Tzu hsiang Chien, Nikolai Chinaev, Yi Chou, Zhaohui Chu, Benjamin Coleman, Pooja Consul, Sam Conway-Rahman, Scott Crowell, Dylan Cutler, Vivek Dani, Samira Daruki, Anil Das, Daniel Deutsch, Nishanth Dikkala, Li Ding, Qiuhan Ding, Shenil Dodhia, Konstantin Donhauser, Tulsee Doshi, Anca Dragan, Alex Druinsky, Sahil Dua, Zoltan Egyed, Danielle Eisenbud, Daniel Eppens, Cindy Fan, Bahare Fatemi, Yassir Fathullah, Vlad Feinberg, Milen Ferev, Sebastian Flennerhag, Takumi Fujimoto, João Gabriel Oliveira, Isaac Galatzer-Levy, João Gante, Simon Geisler, Soham Ghosal, Antonious M. Girgis, Tamara von Glehn, Alec Go, Alhaad Gokhale, Alex Grills, Yiming Gu, Mayank Gupta, Pramod Gupta, Guru Guruganesh, Raia Hadsell, Hamza Harkous, Jitendra Harlalka, Demis Hassabis, Anja Hauth, Joe Heyward, Arian Hosseini, Chih-Yang Hsia, I-Hung Hsu, Xiaopeng Huang, Yangsibo Huang, Kevin Hui, Adrian Hutter, Te I, Fotis Iliopoulos, Advait Jain, Ganesh Jawahar, Ziwei Ji, Qilin Jin, Melvin Johnson, Kandarp Joshi, Arun Kandoor, Wang-Cheng Kang, Koray Kavukcuoglu, Mehran Kazemi, Kathleen Kenealy, Amr Khalifa, Phoebe Kirk, Ivan Korotkov, Suraj Kothawade, Vitaly Kovalev, Neel Kovelamudi, Adam Kraft, Ravin Kumar, Vivek Kumar, Harish Kuppam, Justin Lannin, Chen-Yu Lee, Seungji Lee, Dmitry Lepikhin, Alon Levkovitch, Dongdong Li, Qiujia Li, Valentin Liévin, Ethan Lin, Ziqian Lin, Casper Liu, Tianlin Liu, Tianqi Liu, Xin Liu, Ivan Lobov, Mayank Lunayach, Min Ma, Gagan Madan, Andrii Maksai, Eric Malmi, Michal Matuszak, Daniel McDuff, Gaurav Menghani, Maciej Mikuła, Daniil Mirylenka, Karolis Misiunas, Vedant Misra, Andreea Mitran, Kareem Mohamed, Maksim Mukha, Eric Noland, James

O'Donnell, Brendan O'Donoghue, Kate Olszewska, Bernett Orlando, Wanqiong Pan, Rina Panigrahy, Unnati Parekh, Nicolas Perez-Nieves, Chunjong Park, Eric Paskie, Liqian Peng, Bryce Petrini, Slav Petrov, Jonas Pfeiffer, Bilal Piot, Martyna Plomecka, Siim Poder, Octavio Ponce, Arijit Pramanik, David Racz, Anish Rajan, Michelle Ramanovich, Anand Rao, Marvin Ritter, Vitor Rodrigues, Evan Rosen, Mikołaj Rybiński, Noveen Sachdeva, Michaël E. Sander, Rohit Sathyanarayana, Sagar Savla, Samuel Schmidgall, Tal Schuster, George Scrivener, Benoit Seguin, Andrew Sellergren, Aliaksei Severyn, Izhak Shafran, Dhruv Shah, Bobak Shahriari, Yuan Shangguan, Ashish Shenoy, Pradeep Shenoy, Rakesh Shivanna, Pauline Sho, Lucas Spangher, Wojciech Stokowiec, Tim Strother, Yao Su, Yinghao Sun, Mukund Sundararajan, Andrea Tacchetti, Mor Hazan Taege, Pouya Tafti, Jean Tarbouriech, Chetan Tekur, Shantanu Thakoor, Rahul Thapa, Madeleine Traverse, Lenart Treven, Tao Tu, Chien Te Tung, Çağlar Ünlü, Petar Veličković, Malini Pooni Venkat, Sagar Gubbi Venkatesh, Vidya Venkiteswaran, Francesco Visin, Alex Vitvitskyi, Kiran Vodrahalli, Weiyi Wang, Xin Wang, Tris Warkentin, Jan Wassenberg, John Wieting, Cindy Wu, Lechao Xiao, Hao Xu, Yuhui Xu, Fuzhao Xue, Arun Yadav, Jun Yan, Antoine Yang, Lin Yang, Ming-Hsuan Yang, Ziyu Ying, Jae Hyeon Yoo, Morteza Zadimoghaddam, Sajjad Zafar, Fred Zhang, Jiageng Zhang, Jianyi Zhang, Xiaofan Zhang, Chao Zhao, David Zhou, and Chen Zou. Gemma 4 technical report, 2026. URL https://arxiv.org/abs/2607.02770.

Alperen Tercan and Vinayak S. Prabhu. Thresholded lexicographic ordered multiobjective reinforcement learning. In *ECAI 2024*, pp. 3006–3014, 2024. doi: 10.3233/FAIA240841.

Wen Wang, Jiahua Bao, Yongsiqi Tu, Yihao Liu, Haotian Zhou, Haoxuan Ma, Mengyu Zhou, Wenkui Fan, Junwei He, Xiaoxi Jiang, and Guanjun Jiang. SMOPD: Multi-reward reinforcement learning via specialize-and-merge online policy distillation, 2026. URL https://arxiv.org/abs/2608.03092.

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*, 2026.

Bo Xue, Dake Bu, Ji Cheng, Yuanyu Wan, and Qingfu Zhang. Multi-objective linear reinforcement learning with lexicographic rewards. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 70065–70085. PMLR, 2025. URL https://proceedings.mlr.press/v267/xue25a.html.

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

Zhuolin Yang, Zihan Liu, Yang Chen, Wenliang Dai, Boxin Wang, Sheng-Chieh Lin, Chankyu Lee, Yangyi Chen, Dongfu Jiang, Jiafan He, et al. Nemotron-cascade 2: Post-training llms with cascade rl and multi-domain on-policy distillation. *arXiv preprint arXiv:2603.19220*, 2026.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 5824–5836, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/3fe78a8acf5fda99de95303940a2420c-Abstract.html.

Aohan Zeng, Xin Lv, Zhenyu Hou, Zhengxiao Du, Qinkai Zheng, Bin Chen, Da Yin, Chendi Ge, Chenghua Huang, Chengxing Xie, et al. Glm-5: from vibe coding to agentic engineering. *arXiv preprint arXiv:2602.15763*, 2026.