OMNI-THINKER: Scaling Cross-Domain Generalization in LLMs via Multi-Task RL with Hybrid Rewards

Derek Li [*1] **Jiaming Zhou** [*1] **Amirreza Kazemi** [1] **Qianyi Sun** [1] **Abbas Ghaddar** [1] **Mohammad Ali Alomrani** [1]
Liheng Ma [2] **Yu Luo** [3] **Dong Li** [3] **Feng Wen** [1] **Mark Coates** [2] **Jianye Hao** [3] **Yingxue Zhang** [1]

Abstract

The advancement of general-purpose artificial intelligence relies on large language models (LLMs) that excel across a wide range of tasks, from structured reasoning to creative generation. However, post-training methods like Supervised Fine-Tuning (SFT) often struggle with generalization, favoring memorization over transferable learning. In this work, we introduce OMNI-THINKER, a unified reinforcement learning (RL) framework that enhances LLM performance across diverse tasks by combining rule-based verifiable rewards with generative preference signals via *LLM-as-a-Judge* evaluations. Our approach enables consistent optimization across task types and scales RL-based training to subjective domains. We further investigate training strategies, demonstrating that a curriculum-based progression that orders tasks from structured to open-ended improves performance and reduces forgetting. Experimental results across four domains reveal that curriculum learning improves performance by 5.2% over joint training and 9.1% over model merging. These results highlight the importance of task-aware sampling and hybrid supervision in scaling RL-based post-training for general-purpose LLMs.

1. Introduction

As Large Language Models (LLMs) (Hurst et al., 2024; Liu et al., 2024; Dubey et al., 2024; Yang et al., 2024) evolve into general-purpose agents with applications ranging from creative writing (Marco et al., 2025) to robotics (Gemini-Robotics-Team et al., 2025), the challenge of effective post-training becomes increasingly critical. While Supervised Fine-Tuning (SFT) is the dominant paradigm for adapting LLMs to downstream tasks, it often encourages memorization rather than robust generalization, especially on tasks that differ from the training distribution or require flexible, context-dependent reasoning (Chu et al., 2025).

Reinforcement Learning (RL) has emerged as a promising path to improve generalization, especially in structured tasks like math and coding, where verifiable, rule-based rewards can be effective (DeepSeek-AI et al., 2025; Luo et al., 2025; Kimi-Team et al., 2025). Methods such as Group Relative Policy Optimization (GRPO) have shown that even coarse signals can guide LLMs toward structured, chain-of-thought responses (Shao et al., 2024). However, most RL methods are tailored to deterministic, easily verifiable tasks, limiting their utility in open-ended settings such as question answering and creative writing. Moreover, training LLMs across multiple tasks remains challenging because it requires optimizing for diverse forms of feedback signals, including binary correctness checks in structured tasks and subjective, preference-based judgments in generative ones.

To address this, we propose OMNI-THINKER, a unified RL framework that enables LLMs to learn from both rule-based and generative supervision across tasks. Building on *Reinforcement Learning with Verified Reward (RLVR)*, our method integrates symbolic verifiers with *LLM-as-a-Judge* evaluations (Zheng et al., 2023; Zhang et al., 2025) to handle subjective tasks. We further show that curriculum training, which sequences tasks from structured to open-ended based on *Backward Transfer (BWT)* measurements of task forgettability, significantly improves generalization and reduces forgetting. Across four domains, it achieves average gains of 5.2% and 9.1% over joint training and model merging, respectively.

Our key contributions are: (1) We propose OMNI-THINKER, a unified framework that supports both verifiable and generative supervision under a single policy, scaling RL across four diverse domains. (2) We show that *LLM-as-a-Judge* enables scalable reward signals for open-ended tasks, extending GRPO beyond rule-based domains. (3) We demon-

[*] Equal contribution [1] Huawei Noah's Ark Lab, Montréal, Canada [2] McGill University and Mila - Québec AI Institute [3] Huawei Noah's Ark Lab, Beijing, China. Correspondence to: Derek Li <derek.li2@huawei.com>, Jiaming Zhou <jiaming.zhou@huawei.com>, Yingxue Zhang <yingxue.zhang@huawei.com>.

2nd AI for Math Workshop @ 42nd ICML , Vancouver, Canada, 18 July 2025. Copyright 2025 by the author(s).

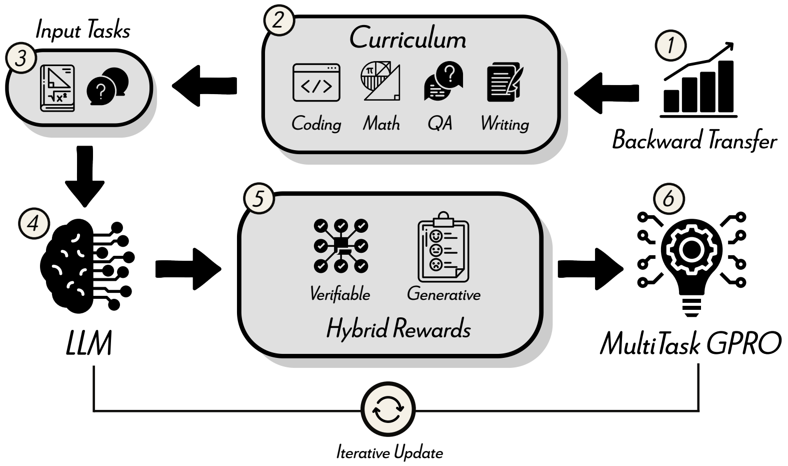

Figure 1. OMNI-THINKER Framework Overview. A unified multi-task RL framework for training a single LLM across structured and open-ended tasks. **(1)** Backward Transfer quantifies task forgettability and informs **(2)** a curriculum that schedules tasks from least to most forgettable. **(3)** Sampled tasks construct prompts. **(4)** The LLM generates completions, which are **(5)** scored by hybrid rewards: verifiable (e.g., test case execution), generative (e.g., LLM-as-a-Judge), and auxiliary (e.g., format adherence), among others. **(6)** The policy is updated via Multi-Task GRPO. The process repeats via iterative updates (bottom loop).

strate that not all multi-task training strategies are equally effective. Curriculum-based task scheduling outperforms uniform sampling and improves generalization in multi-task RL.

2. Related Work

Large Language Models and Multi-Task Learning Early work like (Sanh et al., 2021) showed that multi-task prompted training can encourage zero-shot generalization. Dong et al. (2023) further analyzed how mixing SFT data across domains can cause performance conflicts and forgetting, proposing Dual-stage Mixed Fine-tuning to alleviate these effects. However purely supervised objectives often encourage memorization rather than transferable reasoning. The Qwen3 model series (Yang et al., 2025) employs a four-stage post-training pipeline in the order of reasoning, non-reasoning, and general-domain under a mix of supervised fine-tuning and reinforcement learning. In comparison, the post-training process for Command-A (Cohere et al., 2025) alternates between training multiple expert models separately and merging the experts' parameters into a "Soup Model" during its SFT and RL steps, before the model undergoes a polishing phase of preference alignment. In contrast, our work integrates multi-task learning directly into a single RL framework. Its backward-transfer-guided curriculum orders tasks from least to most forgettable, drawing on continual learning insights (Lopez-Paz & Ranzato, 2017) to reduce interference and maintain stable cross-task performance.

Large Language Models and Reinforcement Learning Reinforcement Learning with Verified Rewards has demonstrated effectiveness for tasks with deterministic correctness signals such as math or code generation (Lambert et al., 2024; Shao et al., 2024; Kimi-Team et al., 2025; Guo et al., 2025). Recent frameworks like General-Reasoner (Ma et al., 2025), Nemotron-Crossthink (Akter et al., 2025) and X-REASONER (Liu et al., 2025) expand this to broader reasoning by blending multi-domain corpora and structured answer templates. However, these tasks still largely remain largely confined to verifiable STEM problems or multiple-choice formats, leaving open-ended generation, such as creative writing, insufficiently addressed. To bridge this gap, Su et al. (2025) propose a generative reward model (GRM) to replace rule-based signals. Although this improves RL and makes it applicable to general-domain QA when references exist, the approach is still restricted to verifiable tasks. In contrast, our approach integrates hybrid verifiable and preference-based rewards within a single RL loop, enabling consistent optimization across both structured and open-ended tasks. Moreover, our curriculum design, guided by backward transfer, helps maintain stable cross-task performance even for tasks lacking deterministic evaluation criteria.

3. Problem Formulation

We consider the problem of training a single language model policy to perform well across a diverse set of tasks, including mathematical reasoning, code generation, question answering, and creative writing. Let $\mathcal{T} = \{T_1, \ldots, T_K\}$ be a collection of K tasks, where each task T_k has a dataset $\mathcal{D}_k$ of input-output pairs (q_k, o_k^*), where $q_k \in \mathcal{Q}_k$ is an input prompt, and $o_k^* \in \mathcal{O}_k^*$ is the corresponding ground-truth output response.

The model, parameterized by θ, defines a conditional distribution $\pi_\theta(y_k \mid x_k)$. Supervised fine-tuning (SFT) aims to maximize the likelihood of ground-truth outputs by minimizing:

$$\min_\theta \sum_{k=1}^{K} \mathbb{E}_{(x_k, y_k) \sim \mathcal{D}_k} \left[\sum_{t=1}^{T} - \log \pi_\theta(o_{k,t}^* \mid q_k, o_{k,<t}^*) \right], \tag{1}$$

where $o_{k,t}^*$ denotes the t-th token of o_k^*, and T is the number of tokens in the output.

While effective in-domain, SFT often overfits to training data and fails to generalize, particularly when task formats vary. Recent studies (Chu et al., 2025) show that reinforcement learning (RL) with outcome-based rewards better promotes cross-task generalization. Motivated by this, we adopt a multi-task RL (MTRL) formulation with task-specific reward functions R_k. The goal is to learn a unified policy π_θ that maximizes the expected reward over the task distribution:

$$\max_\theta \ \mathcal{J}(\theta) = \mathbb{E}_{T_k \sim P(\mathcal{T})} \left[\mathbb{E}_{q_k \sim \mathcal{D}_k, o_k \sim \pi_\theta(\cdot \mid x_k)} \left[R_k(o_k) \right] \right], \tag{2}$$

where $P(\mathcal{T})$ is the task sampling distribution, which determines task exposure during training, and R_k is the task-specific reward function. This objective trains a single policy, π_θ, that balances performance across tasks, enabling knowledge transfer, mitigating negative transfer, and improving generalization across domains.

4. Methodology

4.1. Omni-Thinker Framework Overview

We introduce Omni-Thinker, a unified post-training framework that aligns LLMs to a wide range of tasks via reinforcement learning, extending *RLVR* beyond structured domains like math and code to include general and open-ended tasks.

Verifiable Supervision. For tasks with objective correctness signals, such as symbolic math and code generation, we define binary rewards based on symbolic matches, test case results, or other deterministic evaluators depending on the tasks.

Short-Form Open-Ended Supervision. For language tasks with known or extractable ground-truth answers such as general question answering (QA), we reformulate queries into open-ended prompts and incorporate distractor responses (LLM-generated plausible but incorrect answers) into the context. Instead of labeling options, we prompt the model to reason using the `<think>...</think>` format and to output answers within `<answer>...</answer>` tags. Responses are evaluated with a binary reward based on string matching or set membership against reference answers, thereby encouraging semantic grounding and mitigating shallow pattern memorization. We find that conditioning the LLM on a diverse set of candidate options, including one correct answer and multiple distractors, is key to steadily improving general-domain reasoning while reducing susceptibility to random guessing or reward hacking, compared to directly prompting the model to generate open-ended answers during training without the augmented context.

Long-Form Open-Ended Supervision. For subjective tasks lacking ground truth (e.g., dialogue, writing), we use an *LLM-as-a-Judge* (Chen et al., 2025) to assign a scalar reward based on rubric-aligned pairwise preferences between candidate outputs. This enables learning in domains where symbolic correctness is insufficient or intractable. This prompt-based approach leverages recent advances in the general reasoning capabilities of LLMs, using generated chain-of-thoughts (CoTs) to elicit a ternary reward signal, preferred, tie, or dispreferred, without requiring large-scale preference data collection and reward model training.

Together, these strategies enable consistent RL training across structured and generative tasks within a unified optimization framework.

4.2. Multi-Task Group Relative Policy Optimization

We extend the GRPO (Guo et al., 2025) algorithm to the multi-task setting by jointly optimizing over task-specific reward signals and reference policies. For each input prompt q_k, GRPO samples a group of outputs $\{o_{k,1}, o_{k,2}, \cdots, o_{k,G}\}$ from the old policy $\pi_{\theta_{old}}$. At each generation step t, the partial output is denoted $o_{k,i,<t}$, where $o_{k,i,<t} = [o_{k,i,1}, \ldots, o_{k,i,t-1}]$, that is, the input concatenated with the sequence of generated tokens up to step $t-1$. A reward function $R_k(o, \phi_k)$ scores each output,where ϕ_k denotes task-specific information required for the evaluation, such as reference answers, test cases, or comparison outputs required for evaluation. The policy π_θ is updated to maximize expected return while controlling divergence from a reference policy.

Let

$$\mu_k = \mathrm{mean}\big(\{R_k(o_{k,i}, \phi_k)\}_{i=1}^{G}\big), \quad \sigma_k = \mathrm{std}\big(\{R_k(o_{k,i}, \phi_k)\}_{i=1}^{G}\big), \tag{3}$$

we define the policy ratio $\rho_{k,i,t}$ and the normalized advantage estimate $\hat{A}_{k,i,t}$ as

$$\rho_{k,i,t} = \frac{\pi_\theta(o_{k,i,t} \mid q_k, o_{k,i,<t})}{\pi_{\theta_{\text{old}}}(o_{k,i,t} \mid q_k, o_{k,i,<t})}, \quad \hat{A}_{k,i,t} = \frac{R_k(o_{k,i}, \phi_k) - \mu_k}{\sigma_k}. \tag{4}$$

This allows us to write the MT-GRPO objective as

$$\mathcal{J}_{\text{MT-GRPO}}(\theta) = \mathbb{E}_{k\sim K, q_k\sim\mathcal{D}_k, \{o_{k,i}\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(\cdot|q_k)}$$
$$\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_{k,i}|}\sum_{t=1}^{|o_{k,i}|}\Big\{\min\Big[\rho_{k,i,t}\hat{A}_{k,i}, \text{clip}\left(\rho_{k,i,t}, 1-\epsilon, 1+\epsilon\right)\hat{A}_{k,i}\Big]$$
$$-\beta_k\mathbb{D}_{KL}\left[\pi_\theta||\pi_{ref}\right]\Big\}, \tag{5}$$

where

$$\mathbb{D}_{KL}\left[\pi_\theta||\pi_{ref}\right] = \frac{\pi_{ref}(o_{i,t}|q, o_{i,<t})}{\pi_\theta(o_{i,t}|q, o_{i,<t})} - \log\frac{\pi_{ref}(o_{i,t}|q, o_{i,<t})}{\pi_\theta(o_{i,t}|q, o_{i,<t})} - 1. \tag{6}$$

The clipping parameter ϵ stabilizes updates by keeping policy ratios within a bounded range, following the PPO approach (Schulman et al., 2017). The KL divergence term regularizes the new policy towards the reference policy π_{ref}, weighted by a task-specific coefficient β_k.

4.3. Task Scheduling

We consider two strategies for optimizing a shared policy across multiple tasks: *curriculum learning* and *joint training*. Both utilize task-specific rewards but differ in scheduling. To guide task ordering, we adapt a *Backward Transfer (BWT)* metric to quantify forgetting in our multi-task setting. Following Lopez-Paz & Ranzato (2017), we define $BWT_j = P_{after,j} - P_{base,j}$, where $P_{base,j}$ is the performance on task j from the original base model and $P_{after,j}$ is the performance on task j after training on subsequent tasks. A negative BWT_j indicates that performance on task j degrades due to learning new tasks, measuring the extent of forgetting.

Curriculum Learning: Prior work in curriculum learning prioritizes task complexity (e.g., easy to hard) for more effective learning (Bengio et al., 2009; Parashar et al., 2025). We instead adopt a *forgetting*-based curriculum that orders tasks according to their average forgettability in multi-task learning. Tasks with the least negative or even positive backward transfer are introduced earlier, serving as stable foundations for subsequent tasks that are more prone to catastrophic forgetting. In our setup, this coincides with increasing supervision subjectivity. This *forgetting*-based ordering reduces overall forgetting and improves generalization by minimizing destructive interference across tasks.

Joint Training: All tasks are sampled concurrently from a shared distribution. While this promotes exposure, it introduces three key challenges: (1) simpler tasks may dominate without adaptive sampling due to the tendency of reward hacking in GRPO; (2) reward delays vary, requiring asynchronous updates; and (3) Kullback–Leibler (KL) regularization demands tuning per task due to varying sensitivities.

In both regimes, we apply MT-GRPO with normalized advantages and task-specific KL coefficients. At each step, a task T_k is selected, completions are generated by the current policy π_θ, and updates are applied based on the corresponding reward signal.

5. Experimental Setup

5.1. Training Datasets

We curate a multi-domain training dataset covering Math, Coding, General QA, and Creative Writing, with each domain selected to support hybrid reward functions and robust evaluation. For Math, we begin with the OpenR1-Math (HuggingFace, 2025) dataset, retaining only word problems and excluding questions that require visual reasoning. We further subsample 12,000 examples to fit our compute budget. For Coding, data is sourced from the code-r1-12k (Liu & Zhang, 2025) dataset, with outliers exceeding 1024 tokens removed. Each entry includes a code prompt and JSON-formatted unit tests for automatic validation. For General QA, inspired by SuperGPQA (M-A-P, 2025), we subsample 5,500 queries from the benchmark dataset, proportionally by question category. Each sample comprises a factual question paired with a plain-text answer. We then generate 15 additional confusion options while making sure the uniqueness of correctness by prompting an LLM. The Creative Writing domain leverages 6,650 conversations from Nitral AI's ShareGPT dataset (Nitral-AI, 2024), focused on single-turn completions. Samples exceeding two dialogue turns are filtered out, and responses are judged via an *LLM-as-a-Judge* framework.

5.2. Baselines

We adopt Qwen2.5-7b-Instruct as the base model for all our experiments (Yang et al., 2024). Its robust instruction-following ability makes it a suitable candidate for subsequent reinforcement learning training on both reasoning tasks and more general open-domain QA, as it relies on the model's capacity to comprehend and follow the given prompts effectively.

Supervised Fine-Tuning (SFT): In order to have a meaningful comparison with GRPO, we adopt a similar self-sampled data curation and fine-tuning approach with Rejection sampling Fine-Tuning (Yuan et al., 2023). We first prompt the base model to generate 128 chain-of-thought responses for our training dataset to ensure we end up with at least one correct response for most queries, then filter

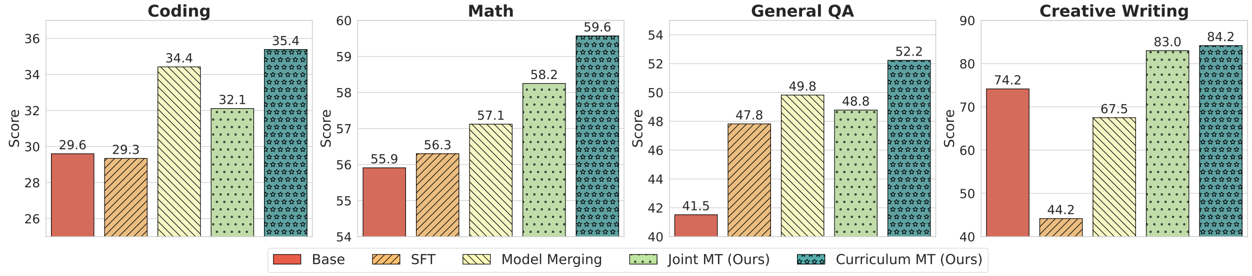

Figure 2. Performance gains across four task domains, comparing our Multi-Task (MT) framework (Joint and Curriculum variants) against baselines including Supervised Fine-Tuning (SFT) and Model Merging. Curriculum MT achieves the strongest results, particularly in open-ended tasks, showing that controlling how multi-task learning is structured is crucial for effective generalization.

them based on the same accuracy reward signals used in GRPO training. We then perform supervised fine-tuning on Qwen2.5-7b-Instruct using these self-distilled responses. This provides a strong on-policy learning baseline that incorporates explicit reasoning steps through self-distillation from the base model.

Model Merging: We employ *TIES-Merging* (Yadav et al., 2023b) as our model-merging baseline. It's a simple yet effective method designed specifically for the multi-task setting that takes into consideration the interference between parameters from models trained on individual tasks during the merging process. It has demonstrated superior performance in multi-task learning compared to linear and task arithmetic approaches (Yadav et al., 2023a). To begin with, we conduct single-task GRPO training using individual task datasets and collect the model weights of the best checkpoints with the help of a validation set for each training run. We then merge the four single-task models using a scaling value $\lambda = 1$.

5.3. Evaluation

We assess performance in each domain using dedicated held-out benchmarks aligned with the task's unique evaluation criteria. The detailed evaluation set is presented as follows

Math Reasoning: we compute accuracy over seven datasets: AIME24 (MAA, 2024), AMC23 (MAA, 2023), Gaokao2023EN (Liao et al., 2024), MATH-500 (Hendrycks et al., 2021), MinervaMath (Lewkowycz et al., 2022), and OlympiadBench (He et al., 2024).

Code Generation: we measure coding ability via pass@1 on BigCodeBench (Complete-Full) (Zhuo et al., 2024) and LiveCodeBench (24Oct-25Jan) (Jain et al., 2024).

General QA: we report exact-match accuracy using the MMLU-Pro benchmark (Wang et al., 2024).

Creative Writing: we evaluate the creative writing task using the *role-play* and *creative writing* subcategory of MT-Bench (Zheng et al., 2023), reporting win rate against a GPT-4 (*pre-gen dated June 16, 2023*) model.

6. Results and Discussion

6.1. Main Results: Scaling Multi-Task LLM Post-Training with OMNI-THINK

We evaluate OMNI-THINKER across four diverse domains: Coding, Math, General QA, and Creative Writing, to assess how reinforcement learning with rule-based verifiable rewards and generative supervision supports multi-task generalization. Figure 2 shows that Curriculum-Guided Multi-Task GRPO consistently yields the best results. Table 3 further details how these gains vary by benchmarks.

In **Math**, Curriculum MT achieves the highest average performance at 59.6%, with the clearest gains on more complex reasoning tasks such as MinervaMath and OlympiadBench. These benchmarks benefit from strong rule-based reward signals and backward-transfer-guided task ordering. In contrast, datasets like AMC23 show minimal change because their relatively high baseline scores likely reflect smaller question sets and potential pretraining overlap rather than robust multi-step problem-solving.

In **General QA**, Curriculum MT again performs best (52.2%), followed by Model Merging (49.8%) and Joint Multi-Task GRPO (48.8%). These improvements are driven by our Short-Form Open-Ended Supervision strategy: instead of generating responses in a fully open-ended and unconstrained fashion, the model is trained to produce complete answer strings given a diverse set of candidate responses, enabling the effective application of verifiable reward through simple string matching when training general-domain tasks.

For **Code Generation**, Curriculum MT achieves 35.4%, slightly ahead of Model Merging. Notably, we only evaluate on the subset of LiveCodeBench(24Oct-25Jan) problems released after Qwen2.5's data cutoff, which ensures that these are unseen test items. This setup highlights Curriculum MT's significant generalization gains on novel problems, explaining the larger improvements on LiveCodeBench relative to static benchmarks like BigCodeBench, where data overlap is more likely.

In **Creative Writing**, the introduction of our Long-Form Open-Ended Supervision strategy, employing the *LLM-as-a-Judge* framework for pairwise evaluation, results in significant performance boosts (Curriculum-Guided at 84.2% and Joint MT at 83.00%), underscoring the advantage of our generative reward approach in subjective, open-ended tasks.

These results support our central hypothesis: The OMNI-THINKER Training Framework, with the help of Curriculum MT, enables a single unified policy to scale across structured and open-ended tasks alike, without relying on interleaving *RLVR* on reasoning tasks and fine-tuning non-reasoning tasks.

6.2. Training Order Matters: From Least to Most Forgettable

Our analysis highlights that task ordering is critical for robust multi-task reinforcement learning. Figure 3 summarizes the average backward transfer (BWT) each task receives when training single-task GRPO models on other tasks. Specifically, the x-axis of Figure 3 represents the target task. A negative average BWT value for a target task indicates that it tends to suffer forgetting or experience a decrease in performance when other tasks are trained before it; a positive value suggests that the target task tends to benefit or experience positive transfer. This high-level view reveals that Creative Writing is the most vulnerable task, as it experiences the most negative average BWT received, while Coding is the most resilient, showing a positive average BWT received.

For example, Coding benefits on average after training on another task (average BWT received $+1.06\%$, indicating a positive transfer). In contrast, Creative Writing is the most vulnerable to forgetting: it has the highest magnitude of negative average BWT received (-4.57%), meaning it easily loses performance when other tasks are trained first.

Figure 5 shows the full BWT matrix, where x-axis consists of target tasks and y-axis represent source tasks that are being trained on. Each cell (row_i, col_j) in the matrix represents the BWT from source task j to target task i. A negative value indicates that training on source task j causes forgetting in target task i, while a positive value suggests positive transfer. We observe that structured tasks like Coding and Math are both less forgetting conducive as sources and tend to receive more positive transfers as targets from training on other tasks.

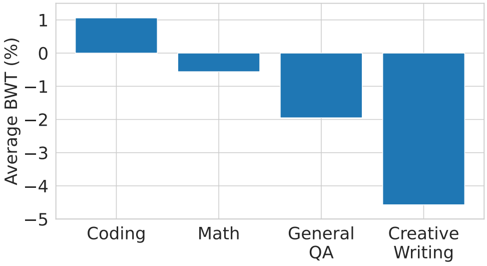

Figure 3. Average *backward transfer (BWT)* per target task, showing how training single-task GRPO on all other tasks affects it on average. A negative BWT indicates forgetting; a positive value indicates positive transfer.

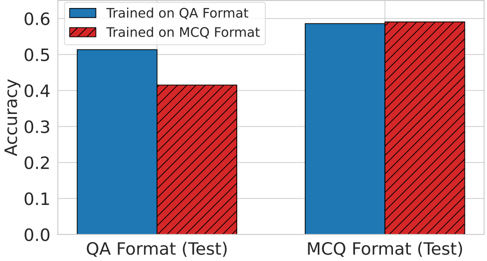

Figure 4. Models trained to generate full-text answers perform better than those trained to select letter choices, especially in free-form evaluation. This training format encourages deeper semantic understanding, rather than relying on shallow pattern matching or guessing.

Guided by this analysis, our curriculum trains tasks in a least-to-most forgettable order: Coding → Math → General QA → Creative Writing. This progression grounds the model in stable, verifiable tasks before introducing those more prone to interference and forgetting. As confirmed by Table 2, reversing this order significantly harms performance.

Overall, our findings suggest that BWT patterns can offer a practical, empirically informed guide for task sequencing, helping reduce negative transfer and improve generalization in multi-task RL for LLMs. Further work is needed to test this approach across a broader range of tasks and domains, including those that require logical reasoning over graph-structured data (Zhou et al., 2024) and knowledge-base retrieval (Dehghan et al., 2024), as well as more diverse open-ended domains such as mixed-initiative collaborative storytelling and co-creativity (Kreminski et al., 2024).

6.3. Output Format Matters: Full-Text Answers Enhance Generalization

We examine how output format impacts generalization by comparing models trained to generate full-text answers versus selecting letter choices in multiple-choice QA (MCQ). Using GRPO, we train two single-task policies on the training set, one prompted to produce full-text final answers at the end of its chain-of-thought completions, and the other to output only letter choices (e.g., “A”, “B”, “C”). As shown in Figure 4, the model trained to output full-text answer achieves significantly better generalization when evaluated with free-form QA prompts on MMLU Pro (51% vs. 41%). While the letter-choice model slightly outperforms when evaluated strictly on MCQ prompts, the full-text model remains competitive across both prompt formats.

These results suggest that training with complete, semantically grounded answers encourages deeper reasoning, improving the model’s ability to generalize beyond the specific format seen during training. In contrast, letter-choice training risks overfitting to shallow pattern matching, reducing transferability to realistic QA settings that often require articulated responses.

7. Conclusion

We presented OMNI-THINKER, a unified framework that extends LLMs to handle both structured and open-ended tasks within a single policy. By integrating reinforcement learning with both rule-based verifiable rewards and generative preference-based supervision, our method improves generalization while addressing challenges such as forgetting and task interference. Our findings show that effective multi-task LLM post-training depends not only on reward design but also on how tasks are sequenced and optimized together. By ordering tasks from structured to open-ended domains based on backward transfer, OMNI-THINKER minimizes forgetting and achieves consistent gains across diverse reasoning and generative tasks. Overall, it takes a step toward more general-purpose LLMs that can learn effectively from both verifiable and subjective feedback, bridging structured reasoning, open-ended question answering, and creative generation within a single, unified post-training framework.

References

Akter, S. N., Prabhumoye, S., Novikov, M., Han, S., Lin, Y., Bakhturina, E., Nyberg, E., Choi, Y., Patwary, M., Shoeybi, M., and Catanzaro, B. Nemotron-crossthink: Scaling self-learning beyond math reasoning, 2025.

Bengio, Y., Louradour, J., Collobert, R., and Weston, J. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pp. 41–48, 2009.

Chen, N., Hu, Z., Zou, Q., Wu, J., Wang, Q., Hooi, B., and He, B. Judgelrm: Large reasoning models as a judge. *arXiv preprint arXiv:2504.00050*, 2025.

Chu, T., Zhai, Y., Yang, J., Tong, S., Xie, S., Schuurmans, D., Le, Q. V., Levine, S., and Ma, Y. Sft memorizes, rl generalizes: A comparative study of foundation model post-training, 2025.

Cohere, T., Ahmadian, A., Ahmed, M., Alammar, J., Alizadeh, M., Alnumay, Y., Althammer, S., Arkhangorodsky, A., Aryabumi, V., Aumiller, D., et al. Command a: An enterprise-ready large language model. *arXiv preprint arXiv:2504.00698*, 2025.

DeepSeek-AI, Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., Xu, R., Zhu, Q., Ma, S., Wang, P., Bi, X., Zhang, X., Yu, X., Wu, Y., Wu, Z. F., Gou, Z., Shao, Z., Li, Z., Gao, Z., Liu, A., Xue, B., Wang, B., Wu, B., Feng, B., Lu, C., Zhao, C., Deng, C., Zhang, C., Ruan, C., Dai, D., Chen, D., Ji, D., Li, E., Lin, F., Dai, F., Luo, F., Hao, G., Chen, G., Li, G., Zhang, H., Bao, H., Xu, H., Wang, H., Ding, H., Xin, H., Gao, H., Qu, H., Li, H., Guo, J., Li, J., Wang, J., Chen, J., Yuan, J., Qiu, J., Li, J., Cai, J. L., Ni, J., Liang, J., Chen, J., Dong, K., Hu, K., Gao, K., Guan, K., Huang, K., Yu, K., Wang, L., Zhang, L., Zhao, L., Wang, L., Zhang, L., Xu, L., Xia, L., Zhang, M., Zhang, M., Tang, M., Li, M., Wang, M., Li, M., Tian, N., Huang, P., Zhang, P., Wang, Q., Chen, Q., Du, Q., Ge, R., Zhang, R., Pan, R., Wang, R., Chen, R. J., Jin, R. L., Chen, R., Lu, S., Zhou, S., Chen, S., Ye, S., Wang, S., Yu, S., Zhou, S., Pan, S., Li, S. S., Zhou, S., Wu, S., Ye, S., Yun, T., Pei, T., Sun, T., Wang, T., Zeng, W., Zhao, W., Liu, W., Liang, W., Gao, W., Yu, W., Zhang, W., Xiao, W. L., An, W., Liu, X., Wang, X., Chen, X., Nie, X., Cheng, X., Liu, X., Xie, X., Liu, X., Yang, X., Li, X., Su, X., Lin, X., Li, X. Q., Jin, X., Shen, X., Chen, X., Sun, X., Wang, X., Song, X., Zhou, X., Wang, X., Shan, X., Li, Y. K., Wang, Y. Q., Wei, Y. X., Zhang, Y., Xu, Y., Li, Y., Zhao, Y., Sun, Y., Wang, Y., Yu, Y., Zhang, Y., Shi, Y., Xiong, Y., He, Y., Piao, Y., Wang, Y., Tan, Y., Ma, Y., Liu, Y., Guo, Y., Ou, Y., Wang, Y., Gong, Y., Zou, Y., He, Y., Xiong, Y., Luo, Y., You, Y., Liu, Y., Zhou, Y., Zhu, Y. X., Xu, Y., Huang, Y., Li, Y., Zheng, Y., Zhu, Y., Ma, Y., Tang, Y., Zha, Y., Yan, Y., Ren, Z. Z., Ren, Z., Sha, Z., Fu, Z., Xu, Z., Xie, Z., Zhang, Z., Hao, Z., Ma, Z., Yan, Z., Wu, Z., Gu, Z., Zhu, Z., Liu, Z., Li, Z., Xie, Z., Song, Z., Pan, Z., Huang, Z., Xu, Z., Zhang, Z., and Zhang, Z. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025.

Dehghan, M., Alomrani, M., Bagga, S., Alfonso-Hermelo, D., Bibi, K., Ghaddar, A., Zhang, Y., Li, X., Hao, J., Liu, Q., et al. Ewek-qa: Enhanced web and efficient knowledge graph retrieval for citation-based question answering systems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14169–14187, 2024.

Dong, G., Yuan, H., Lu, K., Li, C., Xue, M., Liu, D., Wang, W., Yuan, Z., Zhou, C., and Zhou, J. How abilities in large language models are affected by supervised fine-tuning data composition. *arXiv preprint arXiv:2310.05492*, 2023.

Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Yang, A., Fan, A., et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Gemini-Robotics-Team, Abeyruwan, S., Ainslie, J., Alayrac, J.-B., Arenas, M. G., Armstrong, T., Balakrishna, A., Baruch, R., Bauza, M., Blokzijl, M., Bohez, S., Bousmalis, K., Brohan, A., Buschmann, T., Byravan, A., Cabi, S., Caluwaerts, K., Casarini, F., Chang, O., Chen, J. E., Chen, X., Chiang, H.-T. L., Choromanski, K., D'Ambrosio, D., Dasari, S., Davchev, T., Devin, C., Palo, N. D., Ding, T., Dostmohamed, A., Driess, D., Du, Y., Dwibedi, D., Elabd, M., Fantacci, C., Fong, C., Frey, E., Fu, C., Giustina, M., Gopalakrishnan, K., Graesser, L., Hasenclever, L., Heess, N., Hernaez, B., Herzog, A., Hofer, R. A., Humplik, J., Iscen, A., Jacob, M. G., Jain, D., Julian, R., Kalashnikov, D., Karagozler, M. E., Karp, S., Kew, C., Kirkland, J., Kirmani, S., Kuang, Y., Lampe, T., Laurens, A., Leal, I., Lee, A. X., Lee, T.-W. E., Liang, J., Lin, Y., Maddineni, S., Majumdar, A., Michaely, A. H., Moreno, R., Neunert, M., Nori, F., Parada, C., Parisotto, E., Pastor, P., Pooley, A., Rao, K., Reymann, K., Sadigh, D., Saliceti, S., Sanketi, P., Sermanet, P., Shah, D., Sharma, M., Shea, K., Shu, C., Sindhwani, V., Singh, S., Soricut, R., Springenberg, J. T., Sterneck, R., Surdulescu, R., Tan, J., Tompson, J., Vanhoucke, V., Varley, J., Vesom, G., Vezzani, G., Vinyals, O., Wahid, A., Welker, S., Wohlhart, P., Xia, F., Xiao, T., Xie, A., Xie, J., Xu, P., Xu, S., Xu, Y., Xu, Z., Yang, Y., Yao, R., Yaroshenko, S., Yu, W., Yuan, W., Zhang, J., Zhang, T., Zhou, A., and Zhou, Y. Gemini robotics: Bringing ai into the physical world, 2025.

Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., Xu, R., Zhu, Q., Ma, S., Wang, P., Bi, X., et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

He, C., Luo, R., Bai, Y., Hu, S., Thai, Z. L., Shen, J., Hu, J., Han, X., Huang, Y., Zhang, Y., Liu, J., Qi, L., Liu, Z., and Sun, M. OlympiadBench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems, 2024. URL https://arxiv.org/abs/2402.14008.

Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, S., Tang, E., Song, D., and Steinhardt, J. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

HuggingFace. Open r1: A fully open reproduction of deepseek-r1, January 2025. URL https://github.com/huggingface/open-r1.

Hurst, A., Lerer, A., Goucher, A. P., Perelman, A., Ramesh, A., Clark, A., Ostrow, A., Welihinda, A., Hayes, A., Radford, A., et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

Jain, N., Han, K., Gu, A., Li, W.-D., Yan, F., Zhang, T., Wang, S., Solar-Lezama, A., Sen, K., and Stoica, I. Livecodebench: Holistic and contamination free evaluation of large language models for code, 2024. URL https://arxiv.org/abs/2403.07974.

Kimi-Team, Du, A., Gao, B., Xing, B., Jiang, C., Chen, C., Li, C., Xiao, C., Du, C., Liao, C., Tang, C., Wang, C., Zhang, D., Yuan, E., Lu, E., Tang, F., Sung, F., Wei, G., Lai, G., Guo, H., Zhu, H., Ding, H., Hu, H., Yang, H., Zhang, H., Yao, H., Zhao, H., Lu, H., Li, H., Yu, H., Gao, H., Zheng, H., Yuan, H., Chen, J., Guo, J., Su, J., Wang, J., Zhao, J., Zhang, J., Liu, J., Yan, J., Wu, J., Shi, L., Ye, L., Yu, L., Dong, M., Zhang, N., Ma, N., Pan, Q., Gong, Q., Liu, S., Ma, S., Wei, S., Cao, S., Huang, S., Jiang, T., Gao, W., Xiong, W., He, W., Huang, W., Wu, W., He, W., Wei, X., Jia, X., Wu, X., Xu, X., Zu, X., Zhou, X., Pan, X., Charles, Y., Li, Y., Hu, Y., Liu, Y., Chen, Y., Wang, Y., Liu, Y., Qin, Y., Liu, Y., Yang, Y., Bao, Y., Du, Y., Wu, Y., Wang, Y., Zhou, Z., Wang, Z., Li, Z., Zhu, Z., Zhang, Z., Wang, Z., Yang, Z., Huang, Z., Huang, Z., Xu, Z., and Yang, Z. Kimi k1.5: Scaling reinforcement learning with llms, 2025.

Kreminski, M., Chung, J. J. Y., and Dickinson, M. Intent elicitation in mixed-initiative co-creativity. In *IUI Workshops*, 2024.

Lambert, N., Morrison, J., Pyatkin, V., Huang, S., Ivison, H., Brahman, F., Miranda, L. J. V., Liu, A., Dziri, N., Lyu, S., et al. T\" ulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024.

Lewkowycz, A., Andreassen, A. J., Dohan, D., Dyer, E., Michalewski, H., Ramasesh, V. V., Slone, A., Anil, C., Schlag, I., Gutman-Solo, T., Wu, Y., Neyshabur, B., Gur-Ari, G., and Misra, V. Solving quantitative reasoning problems with language models. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=IFXTZERXdM7.

Liao, M., Luo, W., Li, C., Wu, J., and Fan, K. Mario: Math reasoning with code interpreter output–a reproducible pipeline. *arXiv preprint arXiv:2401.08190*, 2024.

Liu, A., Feng, B., Xue, B., Wang, B., Wu, B., Lu, C., Zhao, C., Deng, C., Zhang, C., Ruan, C., et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

Liu, J. and Zhang, L. Code-r1: Reproducing r1 for code with reliable rewards. 2025.

Liu, Q., Zhang, S., Qin, G., Ossowski, T., Gu, Y., Jin, Y., Kiblawi, S., Preston, S., Wei, M., Vozila, P., Naumann, T., and Poon, H. X-reasoner: Towards generalizable reasoning across modalities and domains, 2025.

Lopez-Paz, D. and Ranzato, M. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30, 2017.

Luo, M., Tan, S., Wong, J., Shi, X., Tang, W. Y., Roongta, M., Cai, C., Luo, J., Li, L. E., Popa, R. A., and Stoica, I. Deepscaler: Surpassing o1-preview with a 1.5b model by scaling rl, 2025.

M-A-P. SuperGPQA: Scaling LLM evaluation across 285 graduate disciplines, 2025. URL https://arxiv.org/abs/2502.14739.

Ma, X., Liu, Q., Jiang, D., Zhang, G., Ma, Z., and Chen, W. General-reasoner: Advancing llm reasoning across all domains. *arXiv preprint arXiv:2505.14652*, 2025.

MAA. American mathematics competitions. https://maa.org/student-programs/amc/, 2023.

MAA. American invitational mathematics examination. https://maa.org/maa-invitational-competitions/, 2024.

Marco, G., Rello, L., and Gonzalo, J. Small language models can outperform humans in short creative writing: A study comparing SLMs with humans and LLMs. In *Proceedings of the 31st International Conference on Computational Linguistics*. Association for Computational Linguistics, January 2025.

Nitral-AI. Creative_writing-sharegpt. https://huggingface.co/datasets/Nitral-AI/Creative_Writing-ShareGPT, 2024. Dataset.

Parashar, S., Gui, S., Li, X., Ling, H., Vemuri, S., Olson, B., Li, E., Zhang, Y., Caverlee, J., Kalathil, D., et al. Curriculum reinforcement learning from easy to hard tasks improves llm reasoning. *arXiv preprint arXiv:2506.06632*, 2025.

Sanh, V., Webson, A., Raffel, C., Bach, S. H., Sutawika, L., Alyafeai, Z., Chaffin, A., Stiegler, A., Scao, T. L., Raja, A., et al. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207*, 2021.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y., Wu, Y., et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Su, Y., Yu, D., Song, L., Li, J., Mi, H., Tu, Z., Zhang, M., and Yu, D. Crossing the reward bridge: Expanding rl with verifiable rewards across diverse domains, 2025.

Wang, Y., Ma, X., Zhang, G., Ni, Y., Chandra, A., Guo, S., Ren, W., Arulraj, A., He, X., Jiang, Z., Li, T., Ku, M., Wang, K., Zhuang, A., Fan, R., Yue, X., and Chen, W. MMLU-pro: A more robust and challenging multi-task language understanding benchmark. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. URL https://openreview.net/forum?id=y10DM6R2r3.

Yadav, P., Tam, D., Choshen, L., Raffel, C., and Bansal, M. Ties-merging: Resolving interference when merging models, 2023a. URL https://arxiv.org/abs/2306.01708.

Yadav, P., Tam, D., Choshen, L., Raffel, C. A., and Bansal, M. Ties-merging: Resolving interference when merging models. *Advances in Neural Information Processing Systems*, 36:7093–7115, 2023b.

Yang, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Li, C., Liu, D., Huang, F., Wei, H., et al. Qwen2. 5 technical report. *arXiv e-prints*, pp. arXiv–2412, 2024.

Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Yuan, Z., Yuan, H., Li, C., Dong, G., Lu, K., Tan, C., Zhou, C., and Zhou, J. Scaling relationship on learning mathematical reasoning with large language models. *arXiv preprint arXiv:2308.01825*, 2023.

Zhang, L., Hosseini, A., Bansal, H., Kazemi, M., Kumar, A., and Agarwal, R. Generative verifiers: Reward modeling as next-token prediction. In *The Thirteenth International Conference on Learning Representations*, 2025.

Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E., et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36: 46595–46623, 2023.

Zhou, J., Ghaddar, A., Zhang, G., Ma, L., Hu, Y., Pal, S., Coates, M., Wang, B., Zhang, Y., and Hao, J. Enhancing logical reasoning in large language models through graph-based synthetic data. *arXiv preprint arXiv:2409.12437*, 2024.

Zhuo, T. Y., Vu, M. C., Chim, J., Hu, H., Yu, W., Widyasari, R., Yusuf, I. N. B., Zhan, H., He, J., Paul, I., et al. Bigcodebench: Benchmarking code generation with diverse function calls and complex instructions. *arXiv preprint arXiv:2406.15877*, 2024.

A. Appendix

A.1. Reward Estimation

Omni-Thinker employs a hybrid reward system combineing rule-based correctness (math, code, QA) with preference-based supervision (creative writing) in a unified RL framework. We define task-specific reward functions as $r_k(o, \phi_k)$, where o denotes the model output and ϕ_k represents auxiliary task-specific inputs required for evaluation—such as reference answers, unit tests, or candidate responses for comparison. Each reward function captures domain-relevant correctness criteria, assessing whether o satisfies symbolic constraints, passes execution tests, or is preferred over alternatives under subjective evaluation. While some rewards (e.g., math and code) are strictly deterministic, others, such as LLM-as-a-Judge comparisons, are inherently stochastic but executed at low decoding temperature to ensure stable and consistent supervision. All reward functions are designed to be domain-aware and automatable, supporting scalable reinforcement learning across both structured and generative tasks.

Primary Rewards. Each task employs a tailored correctness criterion:

- **Math:**

$$r_{\texttt{math}}(o, a) = \mathbb{1}\left\{\texttt{verify}_{\texttt{math}}(o_{\texttt{ans}}, a) = \texttt{true}\right\},$$

  where a denotes the ground-truth answer, and $o_{\texttt{ans}}$ is the predicted answer extracted from the <answer> tags in the model output o. Symbolic equivalence between $o_{\texttt{ans}}$ and a is verified using a deterministic parser.

- **Code Generation:**

$$r_{\texttt{code}}(o, \texttt{test\_case}) = \mathbb{1}\Big\{\texttt{exec}(o_{\texttt{ans}}) \models \texttt{unittest}(o_{\texttt{ans}}, \texttt{test\_case})\Big\}$$

  where the generated code $o_{\texttt{ans}}$ is executed in a sandboxed environment and evaluated against the unit tests defined by x; $\models$ indicates logical satisfaction.

- **General QA:**

$$r_{\texttt{qa}}(o, a) = \mathbb{1}\left\{o_{\texttt{ans}} = a\right\}$$

  which returns 1 if the predicted answer matches the ground-truth string exactly.

- **Creative Writing:**

$$r_{\text{writing}}(o, o_{ref}) = \begin{cases} 1.0 & \text{if } o \succ o_{ref} \\ 0.5 & \text{if } o \sim o_{ref} \\ 0.0 & \text{if } o \prec o_{ref} \end{cases}$$

  where $o \succ o_{ref}$ indicates that the model output o is preferred over the reference o_{ref}, $o \prec o_{ref}$ means the reference is preferred, and $o \sim o_{ref}$ denotes a tie. Preferences are judged via pairwise comparison by a fixed *LLM-as-a-Judge* model.

Auxiliary Rewards. To encourage structured outputs, we define formatting-based rewards shared across tasks:

$$r_{\texttt{format}}(o) = \mathbb{1}\left\{\texttt{tags\_valid}(o)\right\}$$

$$r_{\texttt{tags}}(o) = \frac{1}{4} \cdot |\texttt{tags\_present}(o)|$$

Here, tags_valid ensures proper nesting of <think> and <answer> tags, while tags_present counts required structural markers.

Total Reward. We define the total reward as a weighted sum over both primary and auxiliary reward components. Let $\mathcal{F}_k = \{r_k^{(1)}, r_k^{(2)}, \ldots, r_k^{(m)}\}$ denote the set of reward functions associated with task k, where each $r_k^{(j)}$ measures a different aspect of correctness. Given a model output o and its associated evaluation context ϕ_k, the total reward is computed as:

$$\mathcal{R}_k(o, \phi_k) = \sum_{r \in \mathcal{F}_k} w_r \cdot r(o, \phi_k),$$

where $w_r \in [0, 1]$ denotes the task-specific weight for reward component r. If a component reward is undefined, e.g., due to malformed or unparsable output, it is omitted from the sum. Samples with no valid reward components are excluded from policy updates.

A.2. Detailed Hyper-Parameters

We show the hyperparameters used in our training in Table1

Table 1. Training Hyperparameters for All Training Settings. **MT** = Multi-Task RL, **ST** = Single-Task RL (e.g., **ST: Math** = RL trained only on math).

Hyperparameter	**Curr. MT**	**Joint MT**	**ST:Coding**	**ST:Math**	**ST:QA**	**ST:Writing**	**SFT**
Model Configuration							
Max Prompt Length	1024	1024	1024	1024	1024	1024	-
Max Response Length	3072	3072	3072	3072	3072	3072	-
Training Settings							
Train Batch Size	256×6	256×6	256×6	256×6	256×6	256×6	128
Learning Rate	1e-6	1e-6	1e-6	1e-6	1e-6	1e-6	2.5e-6
Learning Scheduler	Constant	Constant	Constant	Constant	Constant	Constant	Cosine
Optimizer	AdamW	AdamW	AdamW	AdamW	AdamW	AdamW	AdamW
Grad Clip	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Max Epoch	3	3	3	3	3	3	3
RL Settings							
KL Beta	0.00	0.02	0.001	0.04	0.04	0.00	-
Clip Ratio Low	0.2	0.2	0.2	0.2	0.2	0.2	-
Clip Ratio High	0.2	0.2	0.2	0.2	0.2	0.2	-
N Rollouts	16	16	16	16	16	16	-
Rollout Temperature	1.0	1.0	1.0	1.0	1.0	1.0	-
Rollout Top-P	1.0	1.0	1.0	1.0	1.0	1.0	-
Rollout Top-K	50	50	50	50	50	50	-
LLM-as-a-Judge Settings							
Model	gpt-4.1-mini	gpt-4.1-mini	-	-	-	gpt-4.1-mini	-
Temperature	0.4	0.4	-	-	-	0.4	-

A.3. Results

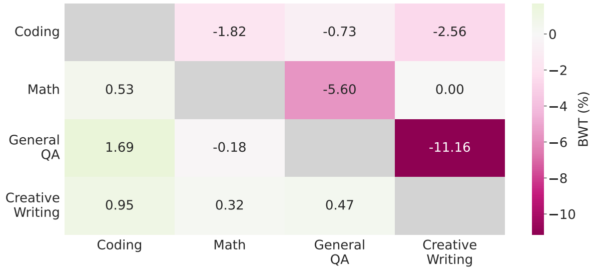

Figure 5. Backward transfer (BWT) matrix. Each cell shows the BWT for a target task (columns) when training a single-task GRPO model on a source task (rows). Positive BWT indicates forgetting; negative indicates positive transfer.

Table 2. Ablation study comparing curriculum Learning (Curr) to its reversed task ordering (Reverse-Curr). Ordering tasks from structured to open-ended domains consistently improves performance across all, while reversing this order leads to degraded results, especially in open-ended domains.

Task	**Base Model**	**Curr**	**Reverse-Curr**
Math	55.9	**59.6**	58.2
General QA	41.5	**52.2**	22.6
Code Generation	29.6	**35.4**	32.7
Creative Writing	74.2	**84.2**	75.0

Table 3. Performance across benchmarks. **MT** = Multi-Task RL, **ST** = Single-Task RL (e.g., **ST: Math** = RL trained only on math). **Model Merge** applies TIES-merging across tasks. Bolded values mark the best per row. Domains include Math (7 sets), MMLU-Pro (9 categories), Coding (2 sets), and Creative Writing (MT-Bench).

Task	**Base Model**	**ST: Coding**	**ST: Math**	**ST: QA**	**ST: Writing**	**SFT**	**Model Merging**	**Joint MT**	**Curriculum MT**
Math									
aime24	15.33	15.33	14.67	14.00	13.33	16.67	14.67	**19.33**	14.67
amc23	62.50	60.00	60.00	62.50	59.00	62.00	**65.50**	64.00	**65.50**
gaokao2023en	72.99	74.29	76.10	74.03	75.58	74.29	74.81	76.62	**77.14**
math500	78.20	78.80	80.40	75.40	79.20	76.80	79.80	77.60	**81.00**
minerva_math	64.34	63.97	66.54	63.24	61.76	65.07	66.18	68.38	**71.69**
OlympiadBench	42.07	42.96	43.70	41.33	42.96	42.96	41.78	43.56	**47.41**
General QA									
Biology	57.60	56.76	52.30	67.36	59.00	66.25	65.55	67.22	**68.76**
Business	33.46	39.04	25.60	58.68	32.95	48.16	**59.82**	49.81	47.53
Chemistry	35.78	31.80	27.30	47.70	38.34	44.08	42.49	42.05	**50.71**
Computer Science	53.66	48.05	50.24	55.12	51.95	53.66	53.90	58.78	**59.27**
Economics	42.65	49.17	38.74	62.91	44.91	59.60	61.97	56.75	**62.09**
Engineering	28.28	31.27	20.43	37.46	26.63	37.77	**38.08**	35.81	37.05
Health	46.70	46.21	45.23	50.98	47.19	45.72	52.69	50.73	**57.09**
History	37.27	33.33	34.65	47.24	38.58	33.86	**47.24**	43.31	45.67
Law	23.16	23.98	20.62	27.88	23.25	26.79	26.61	27.52	**29.70**
Math	55.37	52.63	50.41	59.29	56.25	57.36	58.25	59.22	**61.21**
Other	44.26	40.04	39.72	50.97	43.94	46.43	51.84	49.89	**53.25**
Philosophy	36.87	34.27	33.27	**43.89**	35.47	38.20	41.48	42.08	42.89
Physics	41.11	37.41	30.79	53.66	41.57	49.81	46.73	48.04	**55.58**
Psychology	50.88	51.50	45.36	60.15	51.75	59.02	59.40	59.27	**61.78**
Code Generation									
BigCodeBench	46.49	50.35	46.66	47.10	46.84	44.47	48.07	47.19	**49.47**
LiveCodeBench	12.71	**21.80**	13.07	13.79	13.31	14.21	20.78	17.04	21.30
Creative Writing									
MT-Bench (Writing)	74.16	71.60	74.16	63.00	78.33	44.17	67.50	83.00	**84.17**