Omni-Think: Scaling Multi-Task Learning in LLMs via Reinforcement Learning

Anonymous Authors[1]

Abstract

The pursuit of general-purpose artificial intelligence demands large language models (LLMs) capable of excelling across diverse tasks, ranging from symbolic reasoning to open-ended generation. However, existing post-training methods, such as Supervised Fine-Tuning (SFT) often fall short in multi-task settings, leading to poor generalization and memorization rather than transferable capabilities. In this work, we introduce Omni-Think, a unified framework that enhances LLM performance across both structured and open-ended tasks. Our approach integrates rule-based verifiable rewards with generative preference signals obtained through LLM-as-a-Judge evaluations, enabling consistent optimization across heterogeneous task types. To better understand the dynamics of multi-task RL, we explore different task scheduling strategies and find that introducing tasks in a progression from structured to open-ended leads to better generalization and mitigated forgetting. Experiments across four domains reveals that curriculum training improves average relative performance by 5.2 % over joint multi-task RL and by 9.1 % over merging models trained via RL on individual tasks. These findings highlight the value 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., 2024a; Dubey et al., 2024; Yang et al., 2024) evolve into general-purpose agents, there is growing demand for models that generalize well across a broad range of applications, from creative writing (Marco et al., 2025) to robotics (Team et al., 2025). Yet, current post-training approaches, especially Supervised Fine-Tuning (SFT), often fall short in supporting robust generalization across highly diverse domains (Chu et al., 2025), as they tend to encourage memorization over deep reasoning.

Reinforcement Learning (RL) has recently emerged as a compelling alternative for post-training, particularly in reasoning-intensive domains such as mathematics and coding (DeepSeek-AI et al., 2025; Luo et al., 2025; Kimi-Team et al., 2025). Its success has largely been driven by verifiable rewards: rule-based, often binary signals that provide clear and objective correctness feedback. Notably, Group Relative Policy Optimization (GRPO) (Shao et al., 2024) has shown that even coarse-grained signals can effectively guide LLMs toward structured, chain-of-thought responses.

Nonetheless, existing RL methods have primarily focused on tasks with deterministic, easily verifiable settings. Their applicability to open-ended domains, such as question answering and creative writing, remains limited. Moreover, it is unclear how to perform multi-task preference alignment with heterogeneous reward signals, derived from both verifiable and generative sources, within a unified training paradigm.

Prior efforts have explored two-stage training pipelines, where models are initially trained on reasoning tasks with verifiable rewards, such as those found in math, code, and logical reasoning. In the second stage, these models are further fine-tuned with general reward models to capture human preferences in complex and nuanced scenarios (DeepSeek-AI et al., 2025; Yang et al., 2025). However, general reward models are difficult to train and often suffer from reward hacking, where models exploit flaws in the reward signal instead of genuinely improving output quality (Liu et al., 2024b). Some other recent attempts aim to extend large reasoning models to broader domains (Liu et al., 2025; Ma et al., 2025; Akter et al., 2025), but they typically rely on strict answer formatting for rule-based verification or on a large-scale, high-quality dataset of verifiable tasks to train reward models. These strategies remain limited in their ability to generalize beyond narrow, well-structured settings and struggle to scale to open-ended or subjective domains.

In this paper, we introduce Omni-Think, a unified framework that enhances the performance of LLMs across diverse tasks through reinforcement learning with both verifiable and generative rewards.Our framework combines rule-based verifiers with non-deterministic generative evaluations (Zheng et al., 2023; Zhang et al., 2025). Our approach combines rule-based verifiers with non-deterministic generative evaluations (Zheng et al., 2023; Zhang et al., 2025), thereby enabling RLVR to extend beyond narrowly

defined tasks into subjective or ambiguous domains such as open-domain question answering and creative writing. Additionally, we investigate effective strategies for training LLMs in a multi-task setting. Extensive experiments show that curriculum-based training significantly improves generalization, with average relative gains of 5.2% and 9.1% over naive joint multi-task training and model merging, respectively, across four diverse tasks.

Our key contributions are as follows:

- We propose Omni-Think, a unified training framework that integrates verifiable and generative supervision under a single policy, scaling RL across four diverse domains.
- We demonstrate that LLM-as-a-Judge can be used to convert open-ended tasks into scalable reward signals in a multi-task RL setting, enabling GRPO-style training beyond rule-based domains.
- We show that not all multi-task training strategies are equally effective. Curriculum training, which introduces tasks in a structured progression (e.g., from code to open-ended writing), outperforms joint training with uniform task sampling, resulting in better generalization and reduced forgetting.

2. Problem Formulation

We consider the problem of training a single language model policy to perform well across a diverse set of tasks, such as mathematical reasoning, code generation, question answering, and creative writing. Let $\mathcal{T} = \{T_1, \ldots, T_K\}$ denote a collection of K distinct tasks. Each task T_k is associated with a dataset $\mathcal{D}_k$ containing input-output pairs (x_k, y_k^*), where $x_k \in \mathcal{X}_k$ is an input (e.g., a prompt), and $y_k^* \in \mathcal{Y}_k$ is the corresponding ground-truth output (e.g., response).

The model is parameterized by θ and defines a conditional distribution $\pi_\theta(y \mid x)$ over outputs. The goal of supervised fine-tuning (SFT) is to maximize the likelihood of ground-truth outputs. This is typically achieved by minimizing the following objective:

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

While effective in-domain, SFT often leads to memorization of training data and fails to generalize to out-of-distribution (OOD) settings, especially when tasks vary in rules or modalities. Recent works (Chu et al., 2025) demonstrate that reinforcement learning (RL), particularly with outcome-based rewards, is more effective than SFT in acquiring generalizable knowledge across domains. Motivated by this, we frame our problem under multi-task reinforcement learning (MTRL) with task-specific reward functions (Zeng et al., 2021). For each task T_k, let $R_k : \mathcal{X}_k \times \mathcal{Y}_k \to \mathbb{R}$ denote a scalar reward function that evaluates model outputs.

The objective of MTRL is to learn a unified policy π_θ that maximizes the expected reward across the task distribution. Formally, the RL objective is:

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

where, $P(\mathcal{T})$ is the task sampling distribution, and R_k is the task-specific reward function. The objective is to learn a single policy, π_θ, that performs well on average across a weighted mixture of diverse tasks. The outer expectation is taken over the task sampling distribution $P(\mathcal{T})$, which defines the proportion and order of each task seen during training. The inner expectation is the standard single-task RL objective, which aims to maximize the expected reward for a given task by generating high-quality responses.

By optimizing this unified objective, the model learns to effectively balance performance across all tasks $\mathcal{T}$, enabling knowledge transfer between structured and open-ended domains, mitigate negative transfer, and generalize across reasoning and creative tasks within a single, coherent policy.

3. Methodology

We propose OMNI-THINK, a unified post-training framework for aligning large language models (LLMs) to multiple reasoning tasks via reinforcement learning, scaling RLVR to general and non-reasoning tasks, beyond math reasoning and coding tasks. In addition, we propose a forgetting-aware curriculum for multi-task learning in LLM post-training that optimizes reasoning and non-reasoning tasks in a sequential order.

3.1. Omni-Think: Unified Multi-Task Supervision Scales RLVR Beyond Math

Tasks across diverse domains vary in output structure and supervision format. To support RLVR across such diversity, we introduce the Omni-Think framework, that streamlines multi-task learning especially in non-reasoning tasks with custom reward functions under a shared interface.

Verifiable Supervision. For tasks with well-defined correctness criteria, such as symbolic reasoning or program synthesis, we define $R_k \in \{0, 1\}$ based on symbolic match, test case execution, or other deterministic signals.

Short-Form Open-Ended Supervision. For concise generative tasks with reference answers (e.g., General QA), inspired by Du et al. (2025), we reformulate queries into open-ended prompts and augment with confusion answer choices within the context, and during training we instruct

the model to output in a structured reasoning format and compute binary rewards through string matching or set membership.

During training, we format each input prompt with embedded distractor choices without providing explicit option letters and position indicators. This discourages shallow memorization and instead prompts the model to rely on semantic understanding. The model is thus incentivized to perform internal reasoning using `<think>...</think>` tags before committing to an answer string in the `<answer>...</answer>` segment of its response. This answer string must match one of expected full-text gold answers to receive a positive reward. Such structured prompting aligns closely with the model's chain-of-thought reasoning pathways while still enabling binary verification. Additionally, the reward implementation ensures model's robustness across different query formatting and promotes better transfer to both MCQs and non-MCQ question formats in real-world.

Long-Form Open-Ended Supervision. For tasks with subjective outputs and no ground truth (e.g., creative writing or dialogue), we use a LLM-as-a-Judge (Chen et al., 2025) framework to assess output quality. Given a baseline answer, the evaluator compares the model-generated response against its earlier responses. The resulting reward $R_k \in [0, 1]$ reflects alignment progress with respect to rubric-defined human preferences, enabling learning in domains where exact-match signals are inadequate or unavailable.

This abstraction enables reinforcement learning to operate uniformly across structured and open-ended tasks, providing a consistent optimization objective adaptable to a wide range of supervision modes.

3.2. Multi-Task Learning

We investigate two strategies for optimizing a single policy across diverse reasoning tasks within a unified Multi-Task reinforcement learning framework: (i) *curriculum learning*, and (ii) *joint training*. Both approaches rely on task-specific rewards but differ in how task exposure is structured during training.

Curriculum Learning. Tasks are introduced sequentially in the order: code → math → QA → writing. This ordering reflects an increasing degree of output ambiguity and supervision subjectivity. By beginning with highly structured, verifiable tasks, the model benefits from early stability and inductive scaffolding. Later tasks—such as QA and writing—inherit improved initialization and reward grounding. This curriculum design facilitates transfer across task families and enhances calibration between verifiable and generative objectives.

Joint Training. In contrast, joint training exposes the model to all tasks simultaneously via a shared sampling distribution. Each task maintains its own reward function, and mini-batches are interleaved across domains. While joint training enables broader task exposure, it presents practical challenges: (1) adaptive sampling is required to prevent overfitting to easier tasks; (2) verification time varies across reward types, necessitating asynchronous updates; (3) tasks differ in KL sensitivity, motivating the use of task-specific regularization.

In both strategies, we apply Multi-Task GRPO with task-specific KL penalties and normalized advantages. At each training step, a task T_k is sampled, completions are drawn from the current policy π_θ, and policy updates are computed using the corresponding reward signal.

3.2.1. Multi-Task Group Relative Policy Optimization

We extend the GRPO algorithm to the multi-task setting by jointly optimizing over task-specific reward signals and reference policies. The MT-GRPO objective is defined as:

$$\mathcal{J}_{\text{MT-GRPO}}(\theta) = \mathbb{E}_{k,x_k,\{o_{k,i}\}}\Bigg[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_{k,i}|}\sum_{t=1}^{|o_{k,i}|}\Big(\min\left(r_{k,i,t}\hat{A}_{k,i,t},\ \text{clip}(r_{k,i,t}, 1-\epsilon, 1+\epsilon)\hat{A}_{k,i,t}\right) - \beta_k D_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}})\Big)\Bigg] \quad (3)$$

$$r_{k,i,t} = \frac{\pi_\theta(o_{k,i,t} \mid x_k, o_{k,i,<t})}{\pi_{\theta_{\text{old}}}(o_{k,i,t} \mid x_k, o_{k,i,<t})}, \quad (4)$$

$$\hat{A}_{k,i,t} = \frac{R_k(o_{k,i}) - \mu_k}{\sigma_k}, \quad (5)$$

with $\mu_k = \text{mean}(\{R_k(o_{k,j})\}_{j=1}^{G})$ and σ_k as the corresponding standard deviations. The clipping parameter ϵ follows the PPO-style approach, constraining the policy ratio $r_{k,i,t}$ within a fixed range to prevent overly large updates and stabilize training. KL divergence is regularized against a reference policy π_{ref}, with task-specific coefficients β_k.

4. Experimental Setup

4.1. 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, we extract 5,500 examples from SuperGPQA (M-A-P, 2025), downsampled proportionally by question category. Each sample comprises a factual question paired with a plain-text answer. The Creative Writing domain leverages 6,650 conversations from Nitral AI's ShareGPT dataset(Nitral-AI, 2024), focused on one-turn completions. Samples exceeding two dialogue turns are filtered out, and responses are judged via an LLM-as-a-Judge framework.

4.2. 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), Minerva_Math (Lewkowycz et al., 2022), and OlympiadBench (He et al., 2024).

Code Generation: We measured 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.

Besides, we adapt a *Backward Transfer (BWT)* metric to quantify forgetting in our multi-task setting. We measure BWT as: $BWT_j = P_{base,j} - P_{after,j}$, 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 (Lopez-Paz & Ranzato, 2017). A positive BWT_j indicates forgetting (performance degradation) on task j due to learning new tasks. Specifically, we evaluate how much performance drops on previously learned tasks when the model is trained on new tasks, measuring the extent of catastrophic forgetting in our multi-stage learning scenario.

4.3. 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. Compared to the pretrained Qwen2.5-7B model, using an already competent SFT instruct model provides a strong foundation and helps maximize the additional benefits introduced by our multi-task methodology.

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 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 of 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 dataset 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 $\lambda = 1$.

4.4. Verifiable Reward Design

We define task-specific reward functions $R_k : \mathcal{X} \times \mathcal{Y} \to [0, 1]$, where each function evaluates a model output y given a prompt x with respect to a verifiable notion of correctness. These rewards are designed to be deterministic, domain-aware, and automatable to support scalable reinforcement learning.

Primary Rewards. Each task employs a tailored correctness criterion:

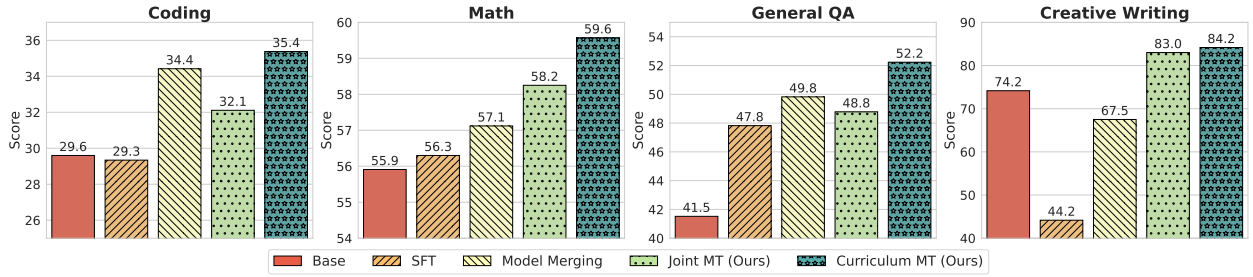

Figure 1. Performance gains across four task domains, comparing our Multi-Task (MT) framework (Joint and Curriculum variants) against baselines including Supervised Fine-Tuning (SFT) and Model Merging. Curriculum MT achieves the strongest results, particularly in open-ended tasks, showing that controlling how multi-task learning is structured is crucial for effective generalization.

- **Math:**

$$R_{\texttt{math}}(x, y) = \mathbb{1}\{\texttt{verify}_{\texttt{math}}(y_{\texttt{ans}}) = \texttt{true}\}$$

where $y_{\texttt{ans}}$ is extracted from <answer> tags, and symbolic equivalence is verified by a deterministic parser.

- **Code Generation:**

$$R_{\texttt{code}}(x, y) = \mathbb{1}\{\texttt{exec}(y_{\texttt{ans}}) \models \texttt{unittest}(x, y_{\texttt{ans}})\}$$

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

- **General QA:**

$$R_{\texttt{qa}}(x, y) = \mathbb{1}\{y_{\texttt{ans}} = y^*_{\texttt{ref}}\}$$

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

- **Creative Writing:**

$$R_{\text{writing}}(x, y) = \begin{cases} 1.0 & \text{if } y \succ y^* \\ 0.5 & \text{if } y \sim y^* \\ 0.0 & \text{if } y \prec y^* \end{cases}$$

where $y \succ y^*$ indicates that the model output y is preferred over the reference y^*, $y \prec y^*$ means the reference is preferred, and $y \sim y^*$ 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:

$$R_{\texttt{format}}(x, y) = \mathbb{1}\{\texttt{tags\_valid}(y)\}$$

$$R_{\texttt{tags}}(x, y) = \frac{1}{4} \cdot |\texttt{tags\_present}(y)|$$

Here, tags_valid ensures proper nesting of <think> and <answer> tags, while tags_present counts required structural markers. Each auxiliary reward is weighted by $w = 0.1$.

Total Reward. Let $\mathcal{R}(x, y) = \sum_{r \in R_k} w_r \cdot r(x, y)$ denote the aggregated reward for a given sample, where $w_r \in [0, 1]$ is a weighting factor for reward component r. If any $r(x, y)$ is undefined (e.g., due to ill-formatted or unparseable output), it is excluded from the sum. Samples for which $\mathcal{R}(x, y)$ is undefined (i.e., zero valid reward components) are filtered prior to policy update.

5. Results and Discussion

5.1. Main Results: Scaling Multi-Task LLM Post-Training with the Omni-Think Framework

To assess the effectiveness of our multi-task RL framework, we conduct experiments across four diverse domains: Coding, Math, General QA, and Creative Writing. Figure 1 highlights the performance improvements, demonstrating that both Curriculum-Guided Multi-Task GRPO consistently outperform the SFT and model merging baselines. Between the two multi-task variants we examine, the curriculum learning design outperforms the joint training paradigm, due in part to easier hyperparameter tuning when training each tasks separately. These results demonstrate that our proposed Omni-Think framework extends RLVR to non-reasoning and creative tasks, allowing us to train a single unified policy to generalize across both structured and open-ended tasks during the same post-training step.

In **Math**, Curriculum MT attains the highest performance (59.6%), notably leveraging structured and verifiable reward signals early in the training. Joint MT similarly outperforms SFT and model merging, validating the Omni-Think approach's effectiveness. 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 Su-

Table 1. Ablation study comparing curriculum Learning (Curr) to its reversed task ordering (Reverse-Curr). Presenting tasks from symbolic reasoning to general reasoning to creative tasks yields consistent performance gains across all domains, 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

pervision 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. This design leads to more robust generalization and aligns well with our multi-task reward formulation. For **Code Generation**, Curriculum MT delivers 35.4%, outperforms both the Model Merging baseline at 34.4% and Joint Multi-Task GRPO with a score of 32.1%, though both of which show improvement upon the SFT and base model. 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%). This clearly underscores the advantage of our generative reward mechanisms over traditional rule-based methods in subjective domains.

These results support our central hypothesis: The Omni-Think 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.

5.1.1. Curriculum Learning Enhances Multi-task Performance

In Figure 1, curriculum-guided training, which introduces tasks in the order of code $\rightarrow$ math $\rightarrow$ QA $\rightarrow$ writing, achieves the strongest performance in all four domains.

To better understand the impact of task ordering, we compare against a reverse curriculum (Table 1). Reversing the task order leads to catastrophic forgetting in General QA (from 52.23 to 22.62), despite moderate retention in Math and Coding abilities.

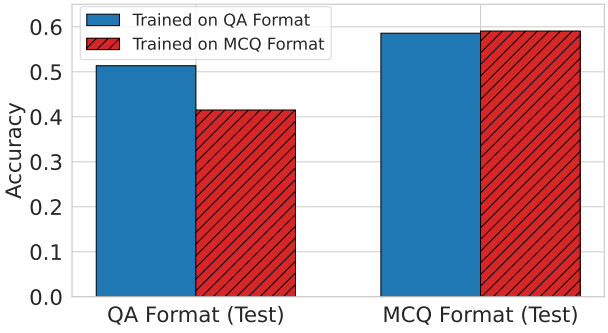

Figure 2. Models trained to generate full-text answers perform better than those trained to select letter choices, especially in free-form evaluation. This training format encourages deeper semantic understanding, rather than relying on shallow pattern matching or guessing.

5.2. Output Format Matters: Letter Choices Undermine Generalization

Our novel approach to improving LLM's general reasoning capability through transforming open-ended QA pairs to Multiple Choice Questions (MCQs) involves requiring the model to generate the full text of the correct answer option during GRPO training. This method was empirically validated against a strategy of training the model to predict only the letter choice. As illustrated in Figure 2, evaluations on MMLU Pro demonstrate the superiority of our full-text generation method for overall generalization.

Specifically, when GRPO models were evaluated by expecting a full-text answer (as shown in the Text Answer (Test) section of Figure 2):

- The model trained on full-text answer generation and reward (blue bar) achieved an average accuracy of 51.35% on MMLU Pro.
- The model trained on letter-choice prediction and reward (red bar) achieved a significantly lower average accuracy of 41.51% on MMLU Pro, indicating its struggle to generalize beyond simple letter prediction.

Conversely, when the same models were evaluated by expecting only a letter choice answer (as shown in the Letter Answer (Test) section of Figure 2):

- The model trained on full-text answer generation and reward (blue bar) achieved an average accuracy of 58.58% on MMLU Pro.
- The model trained on letter-choice prediction and reward (red bar) achieved a slightly higher average accuracy of 59.06% on MMLU Pro. This marginal improvement suggests that while it excels at its specific trained task, its understanding is limited.

These results, supported by Figure 2, indicate that while a model trained specifically for letter-choice prediction performs marginally better when only a letter is expected, it struggles considerably when required to generate the full-text answer. In contrast, the model trained for full-text answer generation performs robustly on both evaluation setups, suggesting that the model actually learns the rationales behind the questions being trained on instead of simply memorizing the options. It is nearly as good at letter-choice prediction ($58.58\% vs. 59.06\%$) and vastly superior at generating the full-text answer ($51.35\% vs. 41.51\%$). This suggests that training for full-text generation encourages a deeper semantic understanding of the question and the answer content, leading to better overall generalization and robustness, rather than learning a superficial mapping to letter choices. This improved generalization is crucial for real-world general reasoning QA tasks where accurate full-text answers are often expected.

5.3. Training Order Matters: From Least to Most Forgettable

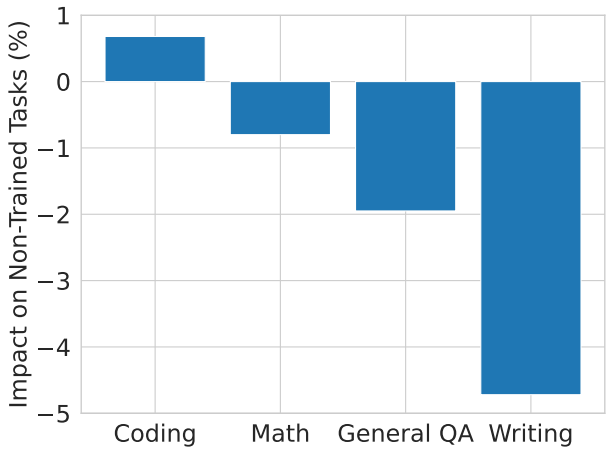

Figure 3. Cross-task backward transfer analysis. Each bar represents the average performance change on the other three tasks when training with single-task GRPO on the source task (x-axis). Coding shows the highest positive transfer, while creative writing leads to the greatest average forgetting.

To minimize forgetting in a multi-stage, multi-task learning setting, the training order of tasks is crucial. Our empirical findings suggest that training should progress from the least forgettable tasks to the most forgettable tasks. This intuition is supported by analyzing the average backward transfer (BWT) across different task types. In our experiment, we trained single-task GRPO models on each domain individually, and measured the degree of forgetting as the performance change on previously learned tasks. Figure 3 demonstrates the average backward transfer (BWT), which shows a clear hierarchy of task forgettability. Coding exhibits the most robust retention, actually showing positive backward transfer (approximately $+0.5\%$) on average. This suggests that coding skills are not only preserved but potentially benefiting through exposure to diverse queries from other tasks, such as a stackoverflow style question in general reasoning. In contrast, Mathematics, General QA, and Creative Writing tasks show negative backward transfer, indicating a forgetting effect when other tasks are introduced. Specifically, Creative Writing experiences the most significant negative transfer (around -4.5%), followed by General QA (around -2%), and Mathematics (less than -1%). Our empirical analysis reveals that task ordering in multi-stage training significantly impacts the degree of catastrophic forgetting across domains.

Based on these findings, our curriculum learning strategy follows the principle of training from least forgettable to tasks easiest to forget: Code $\rightarrow$ Math $\rightarrow$ General QA $\rightarrow$ Creative Writing. This ordering minimizes cumulative forgetting by establishing stable foundations in robust domains before progressing to more fragile knowledge areas. As demonstrated in Table 1, this curriculum-guided approach achieves superior performance compared to reverse ordering, with the reverse curriculum showing catastrophic degradation in General QA (from 52.23% to 22.62%) while our proposed ordering maintains strong performance across all domains, which reinforces the importance of a curriculum-guided training schedule. Starting with highly structured, verifiable tasks like coding and math, which are less susceptible to catastrophic forgetting or even benefit from positive transfer, provides a stable foundation. Subsequent training on more broad tasks like general QA and creative writing then leverages this improved initialization, mitigating the negative backward transfer that these tasks are more prone to.

The implications extend beyond task ordering: these BWT patterns suggest that certain capabilities serve as better "anchors" for multi-task learning, providing stable foundations that support rather than interfere with subsequent learning. This insight informs both the design of training curricula and the selection of foundational capabilities in multi-task language model development.

6. Related Work

Sanh et al. (2021) explore explicit multitask learning to induce zero-shot generalization in language models. They developed a system to map natural language tasks into a human-readable, prompted format, creating a large multi-task mixture with diverse prompts for various supervised datasets. More recently, Dong et al. (2023) investigate how the composition of SFT data influences various abilities in

LLMs, specifically mathematical reasoning, code generation, and general human-aligning abilities.

Reinforcement Learning with Verifiable Rewards (RLVR) (Lambert et al., 2024) has evolved from foundational work in reinforcement learning (RL) and policy optimization. Group Relative Policy Optimization (GRPO) (Shao et al., 2024) was initially introduced in DeepSeekMath as a variant of Proximal Policy Optimization (PPO) (Schulman et al., 2017), simplifying the estimation of the advantage by using group scores from multiple sampled outputs instead of a learned critic model, thereby reducing training resources. Subsequently, Guo et al. (2025) adopted and refined this methodology, notably incorporating a widely successful rule-based reward system into GRPO training. This rule-based reward system primarily focused on accuracy (e.g., exact match for math problems, compiler feedback for code) and format (e.g., specific tags for thinking processes), avoiding learned reward models due to concerns about reward hacking and training complexity.

Several works explore enhancing reasoning across general domains and modalities through multi-task training involving RLVR. Liu et al. (2025) investigate the effects of using supervised finetuning (SFT) and/or reinforcement learning (RL) on general reasoning performance. SFT is performed on general textual domains (e.g, math, medical) with long chain-of-thought (CoT) data generated from a large reasoning model (e.g., o1 (Team, 2024)). RL is also used with verifiable rewards on mathematical textual questions. Their results show that SFT, RL, and their combination can endow models with generalizable reasoning capabilities that transfer across tasks, domains, and modalities despite being trained on textual problems only. However, their RL experiments are limited to mathematical datasets. Ma et al. (2025) propose General-Reasoner, a Zero-RL paradigm that enhances LLM reasoning across diverse domains without intermediate SFT. The authors construct a large-scale, high-quality dataset of verifiable questions for STEM and introduce a compact generative model-based verifier to replace brittle rule-based verification, showing the efficacy of model-based rewards for diverse tasks. Akter et al. (2025) is one of the few works to extend the traditional RL training for LLMs (DeepSeek-AI et al., 2025; Luo et al., 2025) beyond math and coding datasets. The authors identify 2 key steps to perform RL training on broad domains. First, the method unifies answer templates for general domain (i.e., multiple-choice and open-ended) questions to constrain output diversity and make them easily verifiable via a rule-based function. Second, blending multi-domain data using a 2:1 ratio of general-purpose reasoning to math data. The results show improved accuracy and conciseness for all domains which further underscores the importance diverse multi-task training. Su et al. (2025) proposes an alternative RL training framework for general domains by using a model-based reward rather than a rule-based reward. By instructing a generative reward model (GRM) to output a binary score given the candidate response and reference answer, the authors are able to obtain very accurate rewards in general domain scenarios and; hence, increase downstream cross-domain performance.

In addition to the exploration of multi-task post-training by the open-source research community, there have been considerable efforts on the commercial model front. The Qwen3 model series employs a four-stage post-training pipeline (Yang et al., 2025) 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 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 (Cohere et al., 2025).

Different from these works, our method introduces a unified multi-task RL framework that seamlessly integrates both verifiable and generative rewards. Rather than relying solely on rule-based metrics or large external verifiers, we combine lightweight, task-specific reward signals—including LLM-as-a-Judge scoring for open-ended tasks and rule-based verification for structured domains within a single training loop. Our framework enables simultaneous learning across domains and address known challenges in multi-task RL, such as reward imbalance, asynchronous validation latencies, and negative backward transfer(Lopez-Paz & Ranzato, 2017).

7. Conclusion

We present Omni-Think, a unified reinforcement learning framework that enables large language models to learn effectively across a wide range of tasks. By integrating both structured and generative supervision, our method achieves strong, balanced performance across domains. Overall, our findings demonstrate that effective multi-task reinforcement learning depends not only on the reward signals, but also on how tasks are sequenced and optimized together. Omni-Think takes a step toward generalist LLMs capable of adapting to heterogeneous tasks and supervision within a unified training paradigm.

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.

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. URL https://arxiv.org/abs/2501.17161.

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.

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.

Du, X., Yao, Y., Ma, K., Wang, B., Zheng, T., Zhu, K., Liu, M., Liang, Y., Jin, X., Wei, Z., et al. Supergpqa: Scaling llm evaluation across 285 graduate disciplines. *arXiv preprint arXiv:2502.14739*, 2025.

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.

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.

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*, 2024a.

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.

Liu, T., Xiong, W., Ren, J., Chen, L., Wu, J., Joshi, R., Gao, Y., Shen, J., Qin, Z., Yu, T., et al. Rrm: Robust reward model training mitigates reward hacking. *arXiv preprint arXiv:2409.13156*, 2024b.

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.

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.

Team, G. R., 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.

Team, O. Openai o1 system card, 2024. URL https://arxiv.org/abs/2412.16720.

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.

Zeng, S., Anwar, M. A., Doan, T. T., Raychowdhury, A., and Romberg, J. A decentralized policy gradient approach to multi-task reinforcement learning. In *Uncertainty in Artificial Intelligence*, pp. 1002–1012. PMLR, 2021.

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.

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. Results

Table 2. 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: Code	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**